LaTeX 2ε 入门指南

The Not So Short Introduction to LaTeX 2ε (LShort) 中文版

131 分钟学会 LaTeX 2ε

by Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

Version 4.12, 13 April, 2003

翻译： 吴凌云

中文版 4.12， 2005 年 3 月 27 日

Copyright ©1995-2002 Tobias Oetiker and all the Contributers to LShort. All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

中文版权所有 ©2002-2003 CTEX.ORG 以及所有参与 LShort 中文版翻译工作的人员。保留所有权利。

未经授权，不得用于商业目的的出版发行。

中文版前言

LShort 一直是 CTEX 向 LATEX 初学者推荐的入门读物首选。它的优点可以归纳为这么几条：短小的篇幅， 100 多页的内容，不会吓坏初学者；全面的内容，包含了主要的 LATEX 基础知识和大部分常用的 LATEX 命令；丰富的实例，书中使用了大量的例子，通俗易懂；及时的更新， LShort 一直在不停的更新，根据读者的反馈加入新鲜内容、修改错误。 LShort 在世界各地都受到了 LATEX 用户的欢迎，有十几种语言的版本。

LShort 中文版的翻译计划最早是由 CTEX 论坛版主“经典问题”倡议，于 2002 年 5 月开始的。该项目得到了 CTEX 论坛上的众多网友热烈支持。第一个翻译版本基于英文版的 LShort 3.20 。由于大家都是利用业余时间参与，所以翻译工作前后历经将近十个月才得以完成，其间先后多人参与了翻译工作（详见下面列表）。由于各种原因， LShort 3.20 中文版的翻译质量不是很高，很多地方不够清楚，语言也很生硬。而且在 LShort 3.20 中文版翻译结束不久， LShort 就升级到了 4.x 版本。于是我们决定对原来的翻译进行重新修订，并相应的加入英文版 LShort 4.x 版本的新内容。

由于能力所限，书中不可避免还有许多翻译错误或者疏忽，欢迎读者帮助指正，我们将不断修订完善。

吴凌云（ aloft@ctex.org ）

二〇〇五年三月二十七日

以下是曾经参与 LShort 中文版翻译工作的人员名单：

v3.20 经典问题（前言 `overview.tex`），高原之狼（第一章 `things.tex`），controlong（第二章 `typeset.tex`），cxterm（第三章 `math.tex lssym.tex`），aloft（第四章 `spec.tex`），ganzhi（第五章 `custom.tex`）以上均为 CTEX 论坛 ID；

v4.12 吴凌云；

如果你对本书的中文版有任何的建议和意见，请到 CTEX 论坛发表。

CTEX 论坛 http://bbs.ctex.org

本书的最新中文版本可以在以下地址下载
http://project.ctex.org/lshort-cn/

致谢！

本书的部分内容来自于一份由几个奥地利人编写的 LaTeX 2.09 简介（德文版）：

Hubert Partl <partl@mail.boku.ac.at>

Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmwf.ac.at>

Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <noemail>

in Graz

如果你对这份德文文档感兴趣，你可以在 CTAN 中找到由 Jörg Knappen 针对 LaTeX 2_ε 进行更新后的版本：
CTAN:/info/lshort/german

在编写这份文档的时候，我在新闻组 comp.text.tex 上面请求帮助审阅初稿，得到了许多回应。下面这些人帮助进行了校对、提供改进的建议和资料。在他们极其热情的帮助下，这份文档才能得以面世。我对他们所有人表示衷心的感谢。如果书中有什么错误，那都是我自己造成的，而书中每一个正确的地方，甚至每一个单词，都离不开下面这些人的帮助。

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa, Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes, Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey, Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx, Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehradt, Rémi Letot, Johan Lundberg, Alexander Mai, Martin Maechler, Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren, Philipp Nagele, Lenimar Nunes de Andrade, Urs Oswald, Demerson Andre Polli, Maksym Polyakov Hubert Partl, John Refling, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma, Hanspeter Schmid, Craig Schlenter, Baron Schwartz, Christopher Sawtell, Geoffrey Swindale, Laszlo Szathmary, Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone, and Mikhail Zotov.

前言

LaTeX [1] 是一种排版系统，可以生成高质量的科技和数学文档，也能够很好地排版其它各种文档 — 从简单的信函到整本书。LaTeX 使用 TeX [2] 作为它的格式化引擎。

这份简短的入门指南包含了 LaTeX 2_ε 的常用内容，对于日常的 LaTeX 使用应该足够了。更详尽的 LaTeX 使用说明可以参考 [1, 3]。

这份入门指南共分六章：

第一章 介绍 LaTeX 2_ε 文档的基本结构。你还可以了解到一些 LaTeX 的历史。通过阅读本章，你应该对 LaTeX 的工作方式有一个大概的了解。

第二章 深入了解文档的排版过程。本章介绍了大部分基本的 LaTeX 命令和环境。通过阅读本章，你应该能够编写你的第一个文档了。

第三章 了解如何用 LaTeX 排版公式。你可以通过大量的例子来学习 LaTeX 的这一强大功能。本章的最后用表格的形式列出了 LaTeX 中所有的数学符号。

第四章 了解如何使用索引、参考文献和包含 EPS 图形。本章还介绍了如何用 pdfLaTeX 生成 PDF 文档，以及一些常用的扩展宏包。

第五章 介绍 LaTeX 的画图功能。你将学会如何直接使用 LaTeX 来作图，而不是在其他绘图软件中画好再包含到 LaTeX 中。

第六章 介绍一些修改 LaTeX 标准文档布局的方法。这部分内容可能包含有潜在的危险 — 是把 LaTeX 漂亮的输出变得很糟糕还是更加完美就要看你的能力了。

这本书不长，最好能够按顺序阅读这些章节。另外本书提供了丰富的例子，认真阅读这些例子你可以学到很多东西。

LaTeX 在大多数计算机都可以使用，包括个人的 PC 和 Mac 以及大型的 Unix 和 VMS 系统。本书的内容并**不是**告诉你如何安装和设置一个 LaTeX 系统，而是教你如何用 LaTeX 来编写排版你的文档。如果你在安装 LaTeX 系统和设置使用上遇到问题，请询问你的系统管理员或者求助于当地的 LaTeX 组织。

如果你需要任何 LaTeX 相关的资料，不妨访问“ TeX 综合文件网络”（CTAN）的网站。它的主页是 http://www.ctan.org。所有的宏包也可以从 ftp 文件服务器 ftp://www.ctan.org 和遍布全球的各个镜像站点中获得，例如 ftp://ctan.tug.org（美国）， ftp://ftp.dante.de （德国）， ftp://ftp.tex.ac.uk（英国）。如果你不在这些国家中，那么可以在 CTAN 主页上提供的镜像站点名单中选择离你最近服务器。

本书中有很多地方都会提到 CTAN ，特别是提供需要下载的软件和文档的信息的时候。在这些地方没有给出具体的 URL 地址，而是在 CTAN: 之后加上这些内容在 CTAN 目录树中的位置。你需要在前面加上具体的 CTAN 服务器地址以及根目录位置。

如果你想在自己的计算机上安装运行 LaTeX ，可以到 CTAN:/systems 看看有没有合适的系统。

如果你对本书有任何的建议和意见，请告诉我。我对 LaTeX 初学者的反馈特别感兴趣，例如书中哪些部分比较容易理解，而哪些部分应该解释得更清楚些。

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Information Technology and
Electrical Engineering, Swiss Federal Institute of Technology

本书的最新版本可以在以下地址下载
CTAN:/info/lshort

目录

中文版前言 iii

致谢！ v

前言 vii

第一章 Things You Need to Know 1

1.1 The Name of the Game 1

1.1.1 TEX 1

1.1.2 LATEX 2

1.2 Basics 2

1.2.1 Author, Book Designer, and Typesetter 2

1.2.2 Layout Design 3

1.2.3 Advantages and Disadvantages 3

1.3 LATEX Input Files 5

1.3.1 Spaces 5

1.3.2 Special Characters 5

1.3.3 LATEX Commands 6

1.3.4 Comments 7

1.4 Input File Structure 7

1.5 A Typical Command Line Session 8

1.6 The Layout of the Document 10

1.6.1 Document Classes 10

1.6.2 Packages 11

1.6.3 Page Styles 14
1.7 Files You Might Encounter 14
1.8 Big Projects 16

第二章 文档排版 **17**
2.1 文档结构和语言 17
2.2 断行和分页 19
2.2.1 段落调整 19
2.2.2 断字 20
2.3 内建字符串 21
2.4 特殊字符和符号 21
2.4.1 引号 21
2.4.2 破折号和连字号 22
2.4.3 波浪号 (~) 22
2.4.4 度(°) 22
2.4.5 欧洲货币符号() € 23
2.4.6 省略号(...) 24
2.4.7 连字 24
2.4.8 注音符号和特殊符号 24
2.5 多国语言支持 25
2.5.1 对葡萄牙语的支持 27
2.5.2 对法语的支持 29
2.5.3 对德语的支持 30
2.5.4 对韩语的支持 30
2.5.5 对 Cyrillic 的支持 32
2.6 单词的间隔 34
2.7 标题、章和节 34
2.8 交叉引用 36
2.9 脚注 37
2.10 强调 37
2.11 Environments 38

2.11.1 Itemize, Enumerate, and Description 38
2.11.2 Flushleft, Flushright, and Center 39
2.11.3 Quote, Quotation, and Verse 40
2.11.4 逐字打印 40
2.11.5 表格 41
2.12 浮动体 43
2.13 保护脆弱命令 46

第三章 数学公式排版 47
3.1 总论 47
3.2 Grouping in Math Mode 49
3.3 Building Blocks of a Mathematical Formula 50
3.4 Math Spacing 54
3.5 Vertically Aligned Material 55
3.6 Phantoms 57
3.7 Math Font Size 58
3.8 Theorems, Laws, ... 59
3.9 Bold Symbols 60
3.10 List of Mathematical Symbols 61

第四章 特殊功能 71
4.1 插入 Encapsulated PostScript 图形 71
4.2 参考文献 73
4.3 编制索引 75
4.4 定制页眉和页脚 76
4.5 Verbatim 宏包 78
4.6 下载和安装 LaTeX 宏包 78
4.7 使用 pdfLaTeX 79
4.7.1 适于网上发布的 PDF 文件 80
4.7.2 字体 81
4.7.3 使用图形 83
4.7.4 超文本链接 83
4.7.5 链接常见问题 86
4.7.6 书签常见问题 86
4.8 使用 pdfscreen 来演示文档 88

第五章 创建数学图形 **93**
5.1 Overview . . . 93
5.2 picture 环境 . . . 94
5.2.1 基本命令 . . . 94
5.2.2 线段 . . . 96
5.2.3 箭头 . . . 97
5.2.4 画圆 . . . 98
5.2.5 文本和公式 . . . 99
5.2.6 \multiput和\linethickness命令 . . . 100
5.2.7 卵形，\thinlines 和\thicklines 命令 . . . 101
5.2.8 预定义图像块的重复使用 . . . 102
5.2.9 二次Bézier 曲线 . . . 104
5.2.10 悬链线 . . . 105
5.2.11 狭义相对论中的速度 . . . 106
5.3 Xy-pic . . . 106

第六章 Customising LaTeX **111**
6.1 New Commands, Environments and Packages . . . 111
6.1.1 New Commands . . . 112
6.1.2 New Environments . . . 113
6.1.3 Your Own Package . . . 114
6.2 Fonts and Sizes . . . 115
6.2.1 Font Changing Commands . . . 115
6.2.2 Danger, Will Robinson, Danger . . . 118
6.2.3 Advice . . . 118
6.3 Spacing . . . 118
6.3.1 Line Spacing . . . 118
6.3.2 Paragraph Formatting . . . 119
6.3.3 Horizontal Space . . . 120
6.3.4 Vertical Space . . . 121
6.4 Page Layout . . . 122
6.5 More Fun With Lengths . . . 124
6.6 Boxes . . . 125
6.7 Rules and Struts . . . 127

参考文献 129

Index 131

插图

1.1 A Minimal LaTeX File. 8
1.2 Example of a Realistic Journal Article. 9

4.1 fancyhdr 宏包的设置示例。 77
4.2 pdfscreen 输入文件示例 89

6.1 Example Package. 114
6.2 Page Layout Parameters. 123

表格

1.1 Document Classes. 11
1.2 Document Class Options. 12
1.3 Some of the Packages Distributed with LaTeX. 13
1.4 The Predefined Page Styles of LaTeX. 14

2.1 欧洲符号集合 23
2.2 注音符号和特殊符号 25
2.3 Preamble forPortuguese documents. 28
2.4 Special commands for French. 29
2.5 German Special Characters. 30
2.6 Bulgarian, Russian, and Ukrainian 33
2.7 浮动允许放置参量 44

3.1 Math Mode Accents. 61
3.2 Lowercase Greek Letters. 61
3.3 Uppercase Greek Letters. 62
3.4 Binary Relations. 63
3.5 Binary Operators. 64
3.6 BIG Operators. 64
3.7 Arrows. 65
3.8 Delimiters. 65
3.9 Large Delimiters. 65
3.10 Miscellaneous Symbols. 66
3.11 Non-Mathematical Symbols. 66

3.12 AMS Delimiters. 66
3.13 AMS Greek and Hebrew. 66
3.14 AMS Binary Relations. 67
3.15 AMS Arrows. 68
3.16 AMS Negated Binary Relations and Arrows. 69
3.17 AMS Binary Operators. 69
3.18 AMS Miscellaneous. 70
3.19 Math Alphabets. 70

4.1 graphicx 宏包中使用的一些键名 72
4.2 索引关键词格式示意。 76

6.1 Fonts. 116
6.2 Font Sizes. 116
6.3 Absolute Point Sizes in Standard Classes. 116
6.4 Math Fonts. 117
6.5 TEX Units. 121

第一章

Things You Need to Know

The first part of this chapter presents a short overview of the philosophy and history of LaTeX 2ε. The second part focuses on the basic structures of a LaTeX document. After reading this chapter, you should have a rough knowledge of how LaTeX works, which you will need to understand the rest of this book.

1.1 The Name of the Game

1.1.1 TeX

TeX is a computer program created by Donald E. Knuth [2]. It is aimed at typesetting text and mathematical formulae. Knuth started writing the TeX typesetting engine in 1977 to explore the potential of the digital printing equipment that was beginning to infiltrate the publishing industry at that time, especially in the hope that he could reverse the trend of deteriorating typographical quality that he saw affecting his own books and articles. TeX as we use it today was released in 1982, with some slight enhancements added in 1989 to better support 8-bit characters and multiple languages. TeX is renowned for being extremely stable, for running on many different kinds of computers, and for being virtually bug free. The version number of TeX is converging to π and is now at 3.14159.

TeX is pronounced "Tech," with a "ch" as in the German word "Ach" or in the Scottish "Loch." In an ASCII environment, TeX becomes `TeX`.

1.1.2 LaTeX

LaTeX is a macro package that enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout. LaTeX was originally written by Leslie Lamport [1]. It uses the TeX formatter as its typesetting engine. These days LaTeX is maintained by Frank Mittelbach.

LaTeX is pronounced "Lay-tech" or "Lah-tech." If you refer to LaTeX in an `ASCII` environment, you type `LaTeX`. LaTeX 2_ε is pronounced "Lay-tech two e" and typed `LaTeX2e`.

1.2 Basics

1.2.1 Author, Book Designer, and Typesetter

To publish something, authors give their typed manuscript to a publishing company. One of their book designers then decides the layout of the document (column width, fonts, space before and after headings, ...). The book designer writes his instructions into the manuscript and then gives it to a typesetter, who typesets the book according to these instructions.

A human book designer tries to find out what the author had in mind while writing the manuscript. He decides on chapter headings, citations, examples, formulae, etc. based on his professional knowledge and from the contents of the manuscript.

In a LaTeX environment, LaTeX takes the role of the book designer and uses TeX as its typesetter. But LaTeX is "only" a program and therefore needs more guidance. The author has to provide additional information to describe the logical structure of his work. This information is written into the text as "LaTeX commands."

This is quite different from the WYSIWYG[1] approach that most modern word processors, such as *MS Word* or *Corel WordPerfect*, take. With these applications, authors specify the document layout interactively while typing text into the computer. They can see on the screen how the final work will look when it is printed.

[1] What you see is what you get.

When using LaTeX it is not normally possible to see the final output while typing the text, but the final output can be previewed on the screen after processing the file with LaTeX. Then corrections can be made before actually sending the document to the printer.

1.2.2 Layout Design

Typographical design is a craft. Unskilled authors often commit serious formatting errors by assuming that book design is mostly a question of aesthetics—"If a document looks good artistically, it is well designed." But as a document has to be read and not hung up in a picture gallery, the readability and understandability is much more important than the beautiful look of it. Examples:

- The font size and the numbering of headings have to be chosen to make the structure of chapters and sections clear to the reader.
- The line length has to be short enough not to strain the eyes of the reader, while long enough to fill the page beautifully.

With WYSIWYG systems, authors often generate aesthetically pleasing documents with very little or inconsistent structure. LaTeX prevents such formatting errors by forcing the author to declare the *logical* structure of his document. LaTeX then chooses the most suitable layout.

1.2.3 Advantages and Disadvantages

When people from the WYSIWYG world meet people who use LaTeX, they often discuss "the advantages of LaTeX over a normal word processor" or the opposite. The best thing you can do when such a discussion starts is to keep a low profile, since such discussions often get out of hand. But sometimes you cannot escape ...

So here is some ammunition. The main advantages of LaTeX over normal word processors are the following:

- Professionally crafted layouts are available, which make a document really look as if "printed."

- The typesetting of mathematical formulae is supported in a convenient way.
- Users only need to learn a few easy-to-understand commands that specify the logical structure of a document. They almost never need to tinker with the actual layout of the document.
- Even complex structures such as footnotes, references, table of contents, and bibliographies can be generated easily.
- Free add-on packages exist for many typographical tasks not directly supported by basic LaTeX. For example, packages are available to include PostScript graphics or to typeset bibliographies conforming to exact standards. Many of these add-on packages are described in *The LaTeX Companion* [3].
- LaTeX encourages authors to write well-structured texts, because this is how LaTeX works—by specifying structure.
- TeX, the formatting engine of LaTeX 2_ε, is highly portable and free. Therefore the system runs on almost any hardware platform available.

LaTeX also has some disadvantages, and I guess it's a bit difficult for me to find any sensible ones, though I am sure other people can tell you hundreds ;-)

- LaTeX does not work well for people who have sold their souls ...
- Although some parameters can be adjusted within a predefined document layout, the design of a whole new layout is difficult and takes a lot of time.[2]
- It is very hard to write unstructured and disorganized documents.
- Your hamster might, despite some encouraging first steps, never be able to fully grasp the concept of Logical Markup.

[2] Rumour says that this is one of the key elements that will be addressed in the upcoming LaTeX3 system.

1.3 LaTeX Input Files

The input for LaTeX is a plain `ASCII` text file. You can create it with any text editor. It contains the text of the document, as well as the commands that tell LaTeX how to typeset the text.

1.3.1 Spaces

"Whitespace" characters, such as blank or tab, are treated uniformly as "space" by LaTeX. *Several consecutive* whitespace characters are treated as *one* "space." Whitespace at the start of a line is generally ignored, and a single line break is treated as "whitespace."

An empty line between two lines of text defines the end of a paragraph. *Several* empty lines are treated the same as *one* empty line. The text below is an example. On the left hand side is the text from the input file, and on the right hand side is the formatted output.

```
It does not matter whether you
enter one or several     spaces
after a word.

An empty line starts a new
paragraph.
```

It does not matter whether you enter one or several spaces after a word.

An empty line starts a new paragraph.

1.3.2 Special Characters

The following symbols are reserved characters that either have a special meaning under LaTeX or are not available in all the fonts. If you enter them directly in your text, they will normally not print, but rather coerce LaTeX to do things you did not intend.

```
# $ % ^ & _ { } ~ \
```

As you will see, these characters can be used in your documents all the same by adding a prefix backslash:

```
\# \$ \% \^{} \& \_ \{ \} \~{}
```

$ % ^ & _ { } ~

The other symbols and many more can be printed with special commands in mathematical formulae or as accents. The backslash character \ can *not* be entered by adding another backslash in front of it (\\); this sequence is used for line breaking.[3]

1.3.3 LaTeX Commands

LaTeX commands are case sensitive, and take one of the following two formats:

- They start with a backslash \ and then have a name consisting of letters only. Command names are terminated by a space, a number or any other 'non-letter.'
- They consist of a backslash and exactly one non-letter.

LaTeX ignores whitespace after commands. If you want to get a space after a command, you have to put either `{}` and a blank or a special spacing command after the command name. The `{}` stops LaTeX from eating up all the space after the command name.

```
I read that Knuth divides the
people working with \TeX{} into
\TeX{}nicians and \TeX perts.\\
Today is \today.
```

I read that Knuth divides the people working with TeX into TeXnicians and TeXperts.
Today is 2005 年 3 月 27 日.

Some commands need a parameter, which has to be given between curly braces `{ }` after the command name. Some commands support optional parameters, which are added after the command name in square brackets `[ ]`. The next examples use some LaTeX commands. Don't worry about them; they will be explained later.

```
You can \textsl{lean} on me!
```

You can *lean* on me!

```
Please, start a new line
right here!\newline
Thank you!
```

Please, start a new line right here!
Thank you!

[3] Try the `$\backslash$` command instead. It produces a '\'.

1.3.4 Comments

When LaTeX encounters a `%` character while processing an input file, it ignores the rest of the present line, the line break, and all whitespace at the beginning of the next line.

This can be used to write notes into the input file, which will not show up in the printed version.

```
This is an % stupid
% Better: instructive <----
example: Supercal%
              ifragilist%
    icexpialidocious
```

This is an example: Supercalifragilisticexpialidocious

The `%` character can also be used to split long input lines where no whitespace or line breaks are allowed.

For longer comments you could use the `comment` environment provided by the verbatim package. This means, to use the `comment` environment you have to add the command `\usepackage{verbatim}` to the preamble of your document.

```
This is another
\begin{comment}
rather stupid,
but helpful
\end{comment}
example for embedding
comments in your document.
```

This is another example for embedding comments in your document.

Note that this won't work inside complex environments, like math for example.

1.4 Input File Structure

When LaTeX 2_ε processes an input file, it expects it to follow a certain structure. Thus every input file must start with the command

```
\documentclass{...}
```

This specifies what sort of document you intend to write. After that, you can include commands that influence the style of the whole document, or

you can load packages that add new features to the LaTeX system. To load such a package you use the command

```
\usepackage{...}
```

When all the setup work is done,[4] you start the body of the text with the command

```
\begin{document}
```

Now you enter the text mixed with some useful LaTeX commands. At the end of the document you add the

```
\end{document}
```

command, which tells LaTeX to call it a day. Anything that follows this command will be ignored by LaTeX.

Figure 1.1 shows the contents of a minimal LaTeX 2_ε file. A slightly more complicated input file is given in Figure 1.2.

1.5 A Typical Command Line Session

I bet you must be dying to try out the neat small LaTeX input file shown on page 8. Here is some help: LaTeX itself comes without a GUI or fancy buttons to press. It is just a program that crunches away at your input file. Some LaTeX installations feature a graphical front-end where you can click LaTeX into compiling your input file. On other systems there might be some typing involved, so here is how to coax LaTeX into compiling your

[4] The area between `\documentclass` and `\begin{document}` is called the *preamble*.

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

图 1.1: A Minimal LaTeX File.

input file on a text based system. Please note: this description assumes that a working LaTeX installation already sits on your computer.[5]

1. Edit/Create your LaTeX input file. This file must be plain ASCII text. On Unix all the editors will create just that. On Windows you might want to make sure that you save the file in ASCII or *Plain Text* format. When picking a name for your file, make sure it bears the extension `.tex`.

2. Run LaTeX on your input file. If successful you will end up with a `.dvi` file. It may be necessary to run LaTeX several times to get the table of contents and all internal references right. When your input file has a bug LaTeX will tell you about it and stop processing your input file. Type `ctrl-D` to get back to the command line.

   ```
   latex foo.tex
   ```

[5] This is the case with most well groomed Unix Systems, and ... Real Men use Unix, so ... ;-)

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}
```

图 1.2: Example of a Realistic Journal Article.

3. Now you may view the DVI file. There are several ways to do that. You can show the file on screen with

```
xdvi foo.dvi &
```

This only works on Unix with X11. If you are on Windows you might want to try `yap` (yet another previewer).

You can also convert the dvi file to PostScript for printing or viewing with Ghostscript.

```
dvips -Pcmz foo.dvi -o foo.ps
```

If you are lucky your LaTeX system even comes with the `dvipdf` tool, which allows you to convert your `.dvi` files straight into pdf.

```
dvipdf foo.dvi
```

1.6 The Layout of the Document

1.6.1 Document Classes

The first information LaTeX needs to know when processing an input file is the type of document the author wants to create. This is specified with the `\documentclass` command.

`\documentclass[`*options*`]{`*class*`}`

Here *class* specifies the type of document to be created. Table 1.1 lists the document classes explained in this introduction. The LaTeX 2_ε distribution provides additional classes for other documents, including letters and slides. The *options* parameter customises the behaviour of the document class. The options have to be separated by commas. The most common options for the standard document classes are listed in Table 1.2.

Example: An input file for a LaTeX document could start with the line

```
\documentclass[11pt,twoside,a4paper]{article}
```

which instructs LaTeX to typeset the document as an *article* with a base font size of *eleven points*, and to produce a layout suitable for *double sided* printing on *A4 paper*.

1.6.2 Packages

While writing your document, you will probably find that there are some areas where basic LaTeX cannot solve your problem. If you want to include graphics, coloured text or source code from a file into your document, you need to enhance the capabilities of LaTeX. Such enhancements are called packages. Packages are activated with the

```
\usepackage[options]{package}
```

command, where *package* is the name of the package and *options* is a list of keywords that trigger special features in the package. Some packages come with the LaTeX 2ε base distribution (See Table 1.3). Others are provided separately. You may find more information on the packages installed at your site in your *Local Guide* [5]. The prime source for information about LaTeX packages is *The LaTeX Companion* [3]. It contains descriptions on hundreds of packages, along with information of how to write your own extensions to LaTeX 2ε.

表 1.1: Document Classes.

`article`	for articles in scientific journals, presentations, short reports, program documentation, invitations, ...
`report`	for longer reports containing several chapters, small books, PhD theses, ...
`book`	for real books
`slides`	for slides. The class uses big sans serif letters. You might want to consider using FoilTeX[a] instead.

[a] `macros/latex/contrib/supported/foiltex`

表 1.2: Document Class Options.

`10pt`, `11pt`, `12pt` Sets the size of the main font in the document. If no option is specified, `10pt` is assumed.
`a4paper`, `letterpaper`, ... Defines the paper size. The default size is `letterpaper`. Besides that, `a5paper`, `b5paper`, `executivepaper`, and `legalpaper` can be specified.
`fleqn` Typesets displayed formulae left-aligned instead of centred.
`leqno` Places the numbering of formulae on the left hand side instead of the right.
`titlepage`, `notitlepage` Specifies whether a new page should be started after the document title or not. The `article` class does not start a new page by default, while `report` and `book` do.
`onecolumn`, `twocolumn` Instructs LaTeX to typeset the document in one column or two columns.
`twoside, oneside` Specifies whether double or single sided output should be generated. The classes `article` and `report` are single sided and the `book` class is double sided by default. Note that this option concerns the style of the document only. The option `twoside` does *not* tell the printer you use that it should actually make a two-sided printout.
`landscape` Changes the layout of the document to print in landscape mode.
`openright, openany` Makes chapters begin either only on right hand pages or on the next page available. This does not work with the `article` class, as it does not know about chapters. The `report` class by default starts chapters on the next page available and the `book` class starts them on right hand pages.

表 1.3: Some of the Packages Distributed with LaTeX.

doc	Allows the documentation of LaTeX programs. Described in `doc.dtx`[a] and in *The LaTeX Companion* [3].
exscale	Provides scaled versions of the math extension font. Described in `ltexscale.dtx`.
fontenc	Specifies which font encoding LaTeX should use. Described in `ltoutenc.dtx`.
ifthen	Provides commands of the form 'if... then do... otherwise do....' Described in `ifthen.dtx` and *The LaTeX Companion* [3].
latexsym	To access the LaTeX symbol font, you should use the `latexsym` package. Described in `latexsym.dtx` and in *The LaTeX Companion* [3].
makeidx	Provides commands for producing indexes. Described in section 4.3 and in *The LaTeX Companion* [3].
syntonly	Processes a document without typesetting it.
inputenc	Allows the specification of an input encoding such as ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows or user-defined one. Described in `inputenc.dtx`.

[a] This file should be installed on your system, and you should be able to get a `dvi` file by typing `latex doc.dtx` in any directory where you have write permission. The same is true for all the other files mentioned in this table.

1.6.3 Page Styles

LaTeX supports three predefined header/footer combinations—so-called page styles. The *style* parameter of the

`\pagestyle{`*style*`}`

command defines which one to use. Table 1.4 lists the predefined page styles.

表 1.4: The Predefined Page Styles of LaTeX.

`plain`	prints the page numbers on the bottom of the page, in the middle of the footer. This is the default page style.
`headings`	prints the current chapter heading and the page number in the header on each page, while the footer remains empty. (This is the style used in this document)
`empty`	sets both the header and the footer to be empty.

It is possible to change the page style of the current page with the command

`\thispagestyle{`*style*`}`

A description how to create your own headers and footers can be found in *The LaTeX Companion* [3] and in section 4.4 on page 76.

1.7 Files You Might Encounter

When you work with LaTeX you will soon find yourself in a maze of files with various extensions and probably no clue. The following list explains the various file types you might encounter when working with TeX. Please note that this table does not claim to be a complete list of extensions, but if you find one missing that you think is important, please drop me a line.

`.tex` LaTeX or TeX input file. Can be compiled with `latex`.

`.sty` LaTeX Macro package. This is a file you can load into your LaTeX document using the `\usepackage` command.

`.dtx` Documented TeX. This is the main distribution format for LaTeX style files. If you process a .dtx file you get documented macro code of the LaTeX package contained in the .dtx file.

`.ins` The installer for the files contained in the matching .dtx file. If you download a LaTeX package from the net, you will normally get a .dtx and a .ins file. Run LaTeX on the .ins file to unpack the .dtx file.

`.cls` Class files define what your document looks like. They are selected with the `\documentclass` command.

`.fd` Font description file telling LaTeX about new fonts.

The following files are generated when you run LaTeX on your input file:

`.dvi` Device Independent File. This is the main result of a LaTeX compile run. You can look at its content with a DVI previewer program or you can send it to a printer with `dvips` or a similar application.

`.log` Gives a detailed account of what happened during the last compiler run.

`.toc` Stores all your section headers. It gets read in for the next compiler run and is used to produce the table of content.

`.lof` This is like .toc but for the list of figures.

`.lot` And again the same for the list of tables.

`.aux` Another file that transports information from one compiler run to the next. Among other things, the .aux file is used to store information associated with cross-references.

`.idx` If your document contains an index. LaTeX stores all the words that go into the index in this file. Process this file with `makeindex`. Refer to section 4.3 on page 75 for more information on indexing.

`.ind` The processed .idx file, ready for inclusion into your document on the next compile cycle.

`.ilg` Logfile telling what `makeindex` did.

1.8 Big Projects

When working on big documents, you might want to split the input file into several parts. LaTeX has two commands that help you to do that.

`\include{`*filename*`}`

You can use this command in the document body to insert the contents of another file named *filename.tex*. Note that LaTeX will start a new page before processing the material input from *filename.tex*.

The second command can be used in the preamble. It allows you to instruct LaTeX to only input some of the `\included` files.

`\includeonly{`*filename*`,`*filename*`,...}`

After this command is executed in the preamble of the document, only `\include` commands for the filenames that are listed in the argument of the `\includeonly` command will be executed. Note that there must be no spaces between the filenames and the commas.

The `\include` command starts typesetting the included text on a new page. This is helpful when you use `\includeonly`, because the page breaks will not move, even when some included files are omitted. Sometimes this might not be desirable. In this case, you can use the

`\input{`*filename*`}`

command. It simply includes the file specified. No flashy suits, no strings attached.

To make LaTeX quickly check your document you can use the syntonly package. This makes LaTeX skim through your document only checking for proper syntax and usage of the commands, but doesn't produce any (DVI) output. As LaTeX runs faster in this mode you may save yourself valuable time. Usage is very simple:

```
\usepackage{syntonly}
\syntaxonly
```

When you want to produce pages, just comment out the second line (by adding a percent sign).

第二章

文档排版

经过前一章的阅读，对 LaTeX 2ε 文档的基本结构已经有了一个初步的了解。本章将补充一些前面没有涉及到的内容，这样读者就能够生成一个实用的文档了。

2.1 文档结构和语言

By Hanspeter Schmid <hanspi@schmid-werren.ch>

文档写作的主要目的是向读者传递观点、信息或者是知识(一些现代 DAAC [1] 文学除外)。一篇文章如果观点组织得好，读者就能够更好的理解，而只有排版风格切实反映了内容的逻辑和语义结构，读者才能发现和体会到文章的脉络结构。

LaTeX 与其他排版系统的不同在于你只要告诉它文档的逻辑和语义结构，它就可以按照文档类文件(cls)以及其他各种包文件(sty)中定义的规则来生成文档的排版形式。

LaTeX 中最重要的文档单位是段落。之所以称之文档单位是因为段落是反映一个连贯思想或观点的排版形式。下面的部分我们将学到怎样用 \\ 强行断行，以及在源文件中留一空行另起一段。如果开始一个新思想，就应该另起一段，否则，只应使用断行。如果不能确定是否应该另起一段，那就把文档想象为观点和思想的传递者，如果旧的思想还在继续，就不应该另起一段，相反，一段中如果出现全新的观点，就应该另起一段来写。

大多数人完全低估了合理分段的重要性。许多人甚至不知道分段的含义是什么，特别是在 LaTeX 中使用了段落转换而自己却浑然不觉。如果在

[1] 为标新立异不计代价，译自 the Swiss German UVA (Um's Verrecken Anders)。

文档中使用了公式，后者特别容易发生。看看下面的例子，想想为什么有时在方程的前后使用空行(段落转换)，有时却不用。(如果你还不能完全理解这些例子中的命令，请先阅读本章和下一章，再阅读本节内容。)

```
% Example 1
\ldots when Einstein introduced his formula
\begin{equation}
  e = m \cdot c^2 \; ,
\end{equation}
which is at the same time the most widely known
and the least well understood physical formula.


% Example 2
\ldots from which follows Kirchhoff's current law:
\begin{equation}
  \sum_{k=1}^{n} I_k = 0 \; .
\end{equation}

Kirchhoff's voltage law can be derived \ldots


% Example 3
\ldots which has several advantages.

\begin{equation}
  I_D = I_F - I_R
\end{equation}
is the core of a very different transistor model. \ldots
```

下一个较小的文档单位是句子。在英语文档中，句子结尾处句点后的空间大于缩写词句点后的空间。LaTeX 试图分辨出你需要那一个。如果 LaTeX 给错了，你必须告诉 LaTeX 你到底需要什么。在本章的后面将对此予以解释。

文档的结构可以推广到句子。大多数语言具有非常复杂的标点符号系统，但是在很多语言中(包括德语和英语)，只要你记住它表示什么：语言流中的短暂停顿，你就能近乎完美地使用逗号。如果你不能肯定在哪里使用

逗号，就大声地朗读这个句子，在每个逗号处做一短呼吸。在什么地方如果你感到别扭，就删掉这个逗号；如果在什么地方，你感到必须呼吸（或做了短暂停顿）就插入一个逗号。

最后，文档中的段落应该按逻辑组织成更高层次的章、节、子节等形式。但是，使用 `\section{The Structure of Text and Language}` 的排版效果是如此明显，这些高层次结构的使用方法几乎不需要再做说明。

2.2 断行和分页

2.2.1 段落调整

通常书籍是用等长的行来排版的。为了优化整个段落的内容，LaTeX 在单词之间插入必要的断行点（linebreak）和间隔。如果一个单词正好不能完整的排在一行，LaTeX 也会进行必要的断字。段落如何排版依赖于文档类别，通常每一段的第一行有缩进，在两段之间没有额外的间隔。更多的信息请参考第 6.3.2 节。

在某些特殊情形下，必须使用命令让 LaTeX 断行

`\\` or `\newline`

换行而不另起一段。

`\\*`

强行换行，禁止分页。

`\newpage`

另起一页。

`\linebreak[`n`]`, `\nolinebreak[`n`]`, `\pagebreak[`n`]` and `\nopagebreak[`n`]`

上述命令的效果可以从它们的名称看出来。通过可选参数 n，作者可以改变这些命令的效果。n 可以置为 0 和 4 之间的数。如果命令的效果看起来非常差，可以把 n 取为小于 4 的数，让 LaTeX 忽略这个命令，自动作出最好得选择。不要把这些 “break” 命令与 “new” 命令混淆。即使你给出了 “break” 命令，LaTeX 仍然试图使行尾保持平整(通过调整单词间距)或使页面内容充

满整个页面高度(通过调整行间距和段落间距)。如果你真想另起一行，就使用相应的命令。猜猜该是什么命令！

LaTeX 总是尽可能产生最好的断行效果。如果断行无法达到 LaTeX 的高标准，就让这一行在段落的右侧溢出。然后在编译源文件的时候，提示（“overfull hbox”）。这最可能发生在 LaTeX 找不到合适的地方断字的时候。[2]你可以使用 `\sloppy` 命令，告诉 LaTeX 降低一点儿标准。它增加单词之间的间隔，虽然最终的输出结果不是最优，但可以防止出现过长的行。在这种情况下会给出（“underfull hbox”）警告。大多数情况下这样得到的结果看起来不会非常好。`\fussy` 命令可以把 LaTeX 恢复为缺省状态。

2.2.2 断字

LaTeX 在必要时会断字。如果断字算法不能确定正确的断字点，可以使用如下命令告诉 TeX 如何弥补这个缺憾。

命令

`\hyphenation{`*word list*`}`

使列于参量中的单词仅在注有 “-” 的点断字。命令的参量应仅包括由正常字母构成的单词，或 LaTeX 视为正常字母的符号。断字可选点只对使用断字命令时激活的语言起作用。这意味着如果你在文档导言中设置了断字命令，它将影响英文的断字。如果断字命令置于 `\begin{document}` 后面，如果你正使用象 babel 这样的国际语言支持宏包，那么断字可选点在由 babel 激活的语言中就处于活动状态。

下面的例子允许对 “hyphenation” 和 “Hyphenation” 进行断字，而不允许 “FORTRAN”, “Fortran” 和 “fortran” 进行断字。参量中不允许出现特殊的字符和符号。

例子：

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

命令 `\-` 在单词中插入一个自主的断字点。它也就成为这个单词中允许的唯一断字点。对于包含特殊字符（注音字符）的单词，这个命令是特别有用的，因为对于包含特殊字符的单词 LaTeX 不自动断字。

[2]发生（盒子溢出）时，虽然 LaTeX 给出一个警告并显示溢出的那一行，但是不太容易发现溢出的行。如果你在 `\documentclass` 命令中使用选项 `draft`，LaTeX 就在溢出行的右边标以粗黑线。

```
I think this is: su\-per\-cal\-%
i\-frag\-i\-lis\-tic\-ex\-pi\-%
al\-i\-do\-cious
```

I think this is: supercalifragilisticexpialidocious

使用

`\mbox{`*text*`}`

能将几个单词排在同一行上。在任何环境中，这个命令把它的参量排在同一行上。

```
My phone number will change soon.
It will be \mbox{0116 291 2319}.

The parameter
\mbox{\emph{filename}} should
contain the name of the file.
```

My phone number will change soon. It will be 0116 291 2319.

The parameter *filename* should contain the name of the file.

`\fbox` 和 `\mbox` 相似，但是它可以在内容的周围画一个框。

2.3 内建字符串

在前面的例子中，你已经看到用来排特殊字符串的非常简单的 LaTeX 命令。

Command	Example	Description
`\today`	2005 年 3 月 27 日	Current date in the current language
`\TeX`	TeX	The name of your favorite typesetter
`\LaTeX`	LaTeX	The Name of the Game
`\LaTeXe`	LaTeX 2_ε	The current incarnation of LaTeX

2.4 特殊字符和符号

2.4.1 引号

你 不 要像在打字机上那样，把 " 用作引号，印刷中有专门的左引号和右引号。在 LaTeX 中，用两个 ‘ 产生左引号，用两个 ’ 产生右引号。一个 ‘ 和一个 ’ 产生单引号。

```
‘‘Please press the ‘x’ key.’’
```

“Please press the ‘x’ key.”

2.4.2 破折号和连字号

LaTeX 中有四种短划标点符号。连续用不同数目的短划，可以得到其中的三种，第四个不是标点符号，实际是数学中的减号：

```
daughter-in-law, X-rated\\
pages 13--67\\
yes---or no? \\
$0$, $1$ and $-1$
```

daughter-in-law, X-rated
pages 13–67
yes—or no?
0, 1 and −1

这些短划线的名称是: '-' 连字号，'–' 短破折号，'—' 长破折号和 '−' 减号。

2.4.3 波浪号 (~)

波浪号经常和网址用在一起。在 LaTeX 中，可用 \~ 产生，但结果：˜却不是你真正想要的。试一下这个：

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/$\sim$demo
```

http://www.rich.edu/˜bush
http://www.clever.edu/∼demo

2.4.4 度(°)

下面的例子演示如何在 LaTeX 中生成度的符号：

```
It's $-30\,^{\circ}\mathrm{C}$.
I will soon start to
super-conduct.
```

It's −30 °C. I will soon start to super-conduct.

2.4.5 欧洲货币符号() €

写有关钱的文章，需要用到欧洲语言的符号。现在许多字体已经包含了欧洲字符。在文档的导言区加载 textcomp 宏包

```
\usepackage{textcomp}
```

就可以使用命令

```
\texteuro
```

来产生欧洲字符。

如果字体自己不提供欧洲字符，或者是你不喜欢字体中的欧洲符号，可以通过两种办法来解决：

第一，使用 eurosym 宏包。eurosym 宏包提供官方的欧洲符号：

```
\usepackage[official]{eurosym}
```

如果想让欧洲符号和你的字体比较匹配，用 gen 选项替换 official。

如果你的系统中安装了 Adobe 欧洲字体(可以从 ftp://ftp.adobe.com/pub/adobe/type/win/all 免费获得)，就可以使用 europs 宏包、 \EUR 命令(产生与当前字体相匹配的欧洲字符) 或者 eurosans宏包、 \euro 命令(生成官方欧洲符号)。

marvosym 宏包也提供许多种符号，包括欧洲字符，使用 \EUR 命令就可以。缺点是不提供斜体和黑体的欧洲字符。

表 2.1: 欧洲符号集合

package	command	roman	sans-serif	typewriter
eurosym	\euro	€	€	€
[gen]eurosym	\euro	€	€	€
europs	\EUR	€	€	€
eurosans	\euro	€	€	€
marvosym	\EUR	€	€	€

2.4.6 省略号(...)

在打字机上，逗号或句号占据的空间和其他字母相等。在书籍印刷中，这些字符仅占据一点儿空间，并且与前一个字母贴得非常紧，所以不能只键入三个点来输出‘省略号’。有一个专门的命令输出省略号：

```
\ldots
```

```
Not like this ... but like this:\\
New York, Tokyo, Budapest, \ldots
```

Not like this ... but like this:
New York, Tokyo, Budapest, ...

2.4.7 连字

一些字母组合不是简单键入一个个字母得到得的，而实际上用到了一些特殊符号。

ff fi fl ffi... 而不是 ff fi fl ffi ...

这就是所谓的连字，在两个字母之间插入一个 `\mbox{}`，可以禁止连字。对于由两个词构成的单词，这可能是必要的。

```
\Large Not shelfful\\
but shelf\mbox{}ful
```

Not shelfful
but shelfful

2.4.8 注音符号和特殊符号

LaTeX 支持许多语言中的注音符号和特殊字符。表 2.2 就字母 o 列出了所有的注音符号。对于其他字母也自然有效。

在字母 i 和 j 上标一个注音符号，它的点儿必须去掉。这个可使用 `\i` 和 `\j` 生成。

```
H\^otel, na\"\i ve, \'el\`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

2.5 多国语言支持

如果你要写一些英语以外语言的文档，必须要在三个地方对 LaTeX 重新配置：

1. 所有自动生成的字符串[3]必须适当改变以适用于新语言。对于许多种语言，这个问题可由 Johannes Braams 编的宏包 babel 来解决。
2. 对于一种新语言，LaTeX 需要知道它的断字规则。将断字规则输入 LaTeX 有些难度。但是我们可以为不同断字模式重建格式文件，*Local Guide* [5] 给出了更多的信息。
3. 特殊语言的排版规则。例如，在法语中每个冒号(:)前都有一个强制性的空格。

如果你的系统已经配置好了，你可以通过在 `\documentclass` 后使用

`\usepackage[`*language*`]{babel}`

来激活宏包 babel。LaTeX 系统内建的语言列表会在每次编译器启动时显示出来。对于选定的语言，babel 会自动激活适当的断字规则。如果 LaTeX 的 format 文件不支持所选择语言的断字，宏包 babel 仍起作用，只是失去了断字功能，这对于排版效果有很大的负面影响。

[3]表目录, 图目录, ...

表 2.2: 注音符号和特殊符号

ò	`\‘o`	ó	`\’o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`	ç	`\c c`
ǒ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o͟	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`!‘`	¿	`?‘`

宏包 babel 为很多种语言提供专门的新命令来简化特殊字符的输入。例如，German 包含很多元音字符（äöü）。利用 babel，你能用 "o 而不是 \"o 来输入 ö。

要使 babel 宏包使用多种语言：

```
\usepackage[languageA,languageB]{babel}
```

使用

```
\selectlanguage{languageA}
```

来设定当前的语言。

现在大多数的计算机系统允许用户直接从键盘输入多国字符。为了处理不同语系和不同计算机平台的各种输入编码方案，LaTeX 使用 inputenc 宏包：

```
\usepackage[encoding]{inputenc}
```

使用 inputenc 宏包时，应该考虑到或许其他人在其计算机上不能显示你的输入文件，因为他们使用不同的编码。例如，德语字符 ä 在 OS/2 上的编码为 132，但是在一些使用 ISO-LATIN 1 的 Unix 系统上，它的编码为 228，而在 Windows的 cyrillic 语言编码体系 cp1251中，这个字符根本就不存在。所以应小心使用这个功能。根据所使用系统[4]的类型，可能有下列编码类型：

Operating system	encodings	
	western latin	cyrillic
Mac	`applemac`	`macukr`
Unix	`latin1`	`koi8-ru`
Windows	`ansinew`	`cp1251`
DOS, OS/2	`cp850`	`cp866nav`

[4]阅读 `inputenc.dtx` 和 `cyinpenc.dtx` 文档，可以分别对 LaTeX 支持的 Latin 语系和Cyrillic语系编码方案有更多了解。Section 4.6 讲述了如何产生 Package 文档。

如果多语言文档在输入编码上有冲突，我们可以通过 ucs 宏包使用unicode编码。

```
\usepackage{ucs}
\usepackage[utf8]{inputenc}
```

能够以utf8编码创建 LaTeX 源文件。utf8是一种多字节编码方案，每个字符用一个到四个字节来编码。

Font 编码是另外一回事。它的定义是 TeX-字体中每个字符的存储位置。多种输入编码能够影射到同一种font编码，这样就减少了字符集的数量。Font 编码通过fontenc 宏包：

```
\usepackage[encoding]{fontenc}
```

其中，*encoding* 是 font 编码方式。可以同时载入几种编码方式。

缺省的 LaTeX font 编码方法是 OT1，这种编码方法是原始的计算机 Modern TeX 字体，只包括 128 个 7 位 ASCII 字符。当输入注音字母的时候，由 TeX 利用正常字母和注音符号生成。这个方法在对含有注音字母的单词不进行断字时，输出效果很不错。另外，一些 latin 字母是不能通过正常字母和注音符号的组合产生的，更不用说 Greek 或 Cyrillic 这些非 latin 字符了。

为了克服这些缺点，又制造了几种类似与 CM 字体的 8 位字体集。T1 编码中 *Extended Cork* (EC) 字体包含了大多数 Latin语系的字母和标点符号，LH 字体集包含了排版 Cyrillic语系文档的大部分必须字母。由于 Cyrillic 中有大量的图象字符，就分成了四个 font 编码—T2A，T2B，T2C，和 X2。[5]CB子集包含了 LGR 编码中的字体，用来编辑 Greek 文档。

使用这些字体，能够提高/使用非英语文档中的断字能力，新的类 CM 字体的另外一个优点是他们提供了各种形状，各种大小的 CM 字体。

2.5.1 对葡萄牙语的支持

By Demerson Andre Polli <polli@linux.ime.usp.br>

为了正常断字和将自动产生的各种文本变为 葡萄牙语，使用：

```
\usepackage[portuguese]{babel}
```

如果使用的是巴西语，只需要把语言类别替换为 brazilian 即可。

[5]每个编码所支持的语言列表可以在 [11] 中找到。

表 2.3: Preamble forPortuguese documents.

```
\usepackage[portugese]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

葡萄牙语中有许多注音符号，所以你可能要用

```
\usepackage[latin1]{inputenc}
```

来正确的输入这些符号，用

```
\usepackage[T1]{fontenc}
```

来正确的断字。

表 2.3 是你需要在葡萄牙语文档的导言区加入的内容。值得注意的是这里使用的是 latin1 输入编码方案，在 Mac 和 Dos 系统中就不能正常工作了，根据你自己的系统选择合适的编码。

2.5.2 对法语的支持

By Daniel Flipo <daniel.flipo@univ-lille1.fr>

这里是一些提示，对用 LaTeX 编辑 法语 文档的读者会有所帮助：用

```
\usepackage[frenchb]{babel}
```

来载入法语语言支持。

由于历史原因，babel 宏包的法语选项的名字不是 *french*，而是 *frenchb* 或者 *francais*。

如果相应的配置了 LaTeX 系统，就可以实现法语的断字，自动产生的文本也变成法语了，`\chapter` 变为 Chapitre，`\today` 变为法语的当前日期等等。此外还提供了一些新的命令，可以方便的书写法语源文件。表 2.4可能会对你有些启发。

表 2.4: Special commands for French.

`\og guillemets \fg{}`	« guillemets »
`M\up{me}, D\up{r}`	M$^{\text{me}}$, D$^{\text{r}}$
`1\ier{}, 1\iere{}, 1\ieres{}`	1$^{\text{er}}$, 1$^{\text{re}}$, 1$^{\text{res}}$
`2\ieme{} 4\iemes{}`	2$^{\text{e}}$ 4$^{\text{es}}$
`\No 1, \no 2`	N$^{\text{o}}$ 1, n$^{\text{o}}$ 2
`20~\degres C, 45\degres`	20 °C, 45°
`\bsc{M. Durand}`	M. Durand
`\nombre{1234,56789}`	1 234,567 89

在切换成法语后，列表内容的排版结果就会发生变化。有关 babel 宏包 `frenchb` 选项更多的内容，以及它的定制方式，可以阅读 `frenchb.dvi` 文件，这个文件是LaTeX对 `frenchb.dtx` 编译后产生的。

2.5.3 对德语的支持

生成 德语 文档的一些提示：用

```
\usepackage[german]{babel}
```

载入对德语的支持。

同样，在你正确配置 LaTeX 系统后，能够进行断字，将自动产生的文本用德语来表示。例如，“Chapter” 变成了 “Kapitel” 。还提供了一套新命令，即使在不使用 inputenc 宏包的情况下，输入德语文件也更加迅速。表 2.5 可能会有些启发。当使用 inputenc 宏包的时候，这些就成了没有实际作用的东西了，但是你的文件也被局限在某些特定的编码范围里了。

表 2.5: German Special Characters.

`"a`	ä	`"s`	ß
`"‘`	„	`"’`	“
`"< or \flqq`	«	`"> or \frqq`	»
`\flq`	‹	`\frq`	›
`\dq`	"		

德语的书中，经常能够发现法语的引号(«guillemets»)，但是德国的打字工人不是这样用的。正统的德语书中引号是 » 这样子的 «。只有在瑞士一些讲德语的地区的打字员才会使用和法语相同的 « 引号 »。

使用 `\flq` 这样的命令会出现一个问题：如果你的字体是 OT1 (缺省字体)，引号看起来象数学符号：“≪”，打字员是很讨厌这种符号的。T1 字体，包含了所要的符号，如果要使用这种引号，要先确认是使用 T1字体。(`\usepackage[T1]{fontenc}`)

2.5.4 对韩语的支持[6]

用 LaTeX 排版 韩语，需要解决三个问题：

[6]包括了许多韩国 LaTeX 用户必须面对的问题。本部分是由 Karnes KIM 代表 the Korean lshort translation team写成的。由 SHIN Jungshik 翻译成英语，Tobi Oetiker 整理。

1. 我们要能够编辑韩语文件。韩语源文件也纯文本格式，但是 US-ASCII 集之外，韩语使用的是自己的字符集，在一般的 ASCII 编辑器里看起来会是乱码。韩语文件最常用的两种编码方式是 EUC-KR 以及韩语 MS-Windows 使用的它的前向兼容扩展形式， CP949/Windows-949/UHC。在这些编码方式中，每个 US-ASCII 字符和其他象 ISO-8859-x、EUC-JP、Shift_JIS 和 Big5 等 ASCII 兼容编码编码方式一样，表示的是正常的 ASCII 字符。另外，Hangul syllables、Hanjas (韩语中的汉字字符)、Hangul Jamos、Hirakanas、Katakanas、Greek、Cyrillic 字符和其他从 KS X 1001 演化而来的符号使用连续两个 8 进制数表示。现在三种主流操作系统（Mac OS，Unix，Windows）都有相当好的多国语言支持，所以即使在非韩语的操作系统上，编辑韩语文件不再是问题了。

2. TEX 和 LATEX 最初都是针对排版不超过 256 个字母的字符集产生的。为了使他们在具有更多字母的语言中正常工作，比如汉语或韩语，开发了一套子字体技术。它能够把一个 CJK 字体中成千上万的图象字符分割成许多每个具有 256 个字符的子字体。对于韩语，有三种广泛使用的宏包，HLATEX （作者 UN Koaunghi ），hLATEXp （作者 CHA Jaechoon ），以及 CJK 宏包 （作者 Werner Lemberg） 。[7] HLATEX 和 hLATEXp 是特别用来处理韩语的，在字体支持的情况下能够实现韩语的本地化。他们能够处理 EUC-KR 编码的源文件。HLATEX 甚至在和 Λ, Ω 一起使用的时候，能够处理 CP949/Windows-949/UHC 和 UTF-8 编码的文件。

   CJK 宏包并不是专门针对韩语的。它还能够处理 UTF-8 及包括 EUC-KR 和 CP949/Windows -949/UHC 在内的各种 CJK 编码方式的文件，还能够排版多语言文档（特别是汉语，日语和韩语）。CJK 包不象 HLATEX 那样，不提供韩语本地化，也不包括 HLATEX 中所包含的许多韩语字体。

3. 使用 TEX 和 LATEX 这样排版程序的最终目的是为了得到漂亮的文档排版结果。似乎排版中最重要的是一套优秀的字体。HLATEX 发行版包括了 10 种不同族的 UHC PostScript 字体，及 HLATEX 早期版本使用的 5 种不同族的 Munhwabu[8] 字体（TrueType），也可以使用 Bitstream 的 cyberbit TrueType 字体。

[7] 这些宏包能够从 `language/korean/HLaTeX/` `language/korean/CJK/` 和`http://knot.kaist.ac.kr/htex/` 得到。

[8] Korean Ministry of Culture

使用 HLaTeX 宏包排版韩语文档时，将下面的说明放在导言区：

```
\usepackage{hangul}
```

这个命令打开韩语本地化支持。章、节、小节的页眉，目录和图形列表都变成了韩文，文档格式也变成了韩语的习惯。这个包还提供自动“词缀选择”。韩语中有一些后置词缀对在语法上相同，而形式不同。一对中的哪个是正确的要根据前面的音节是母音还是子音来判断（实际上比这复杂，这里只是简单的描述）。以韩语为母语的人是不会弄错词缀的，但是参考文献和其他自动产生的文本就不能确定使用哪个词缀了。每次改变参考文献或者只是简单的移动文档的一部分都手动调整词缀的话，显然是一项辛苦的工作。HLaTeX 可以把用户从这项乏味而且容易出错的工作中解脱出来。

如果不需要韩语的本地化特性，只是排版韩语文本，只需要在导言区输入

```
\usepackage{hfont}
```

用 HLaTeX 排版韩语文件的更详细内容请参考 *HLaTeX 指南*，浏览韩国 TeX 用户组 (KTUG) 的网页 http://www.ktug.or.kr/，有这个手册的韩译本。

2.5.5 对 Cyrillic 的支持

By Maksym Polyakov <polyama@myrealbox.com>

3.7h 版的 babel 包括了对 T2* 编码的支持，可以用 Cyrillic 字母排版 Bulgarian，Russian 和 Ukrainian 文本。

Cyrillic 是标准的 LaTeX 机制通过 fontenc 和 inputenc 宏包来支持的。如果要在数学模式下使用 Cyrillic ，需要在 fontenc 之前加载 mathtext 宏包：[9]

```
\usepackage{mathtext}
\usepackage[T1,T2A]{fontenc}
\usepackage[koi8-ru]{inputenc}
\usepackage[english,bulgarian,russian,ukranian]{babel}
```

一般情况下，babel 宏包会自动选择缺省的字体编码，对于上面三种语言来说是 T2A 编码。但是，文档并不局限于一种字体编码，在使用 Cyrillic

[9] 如果使用 $\mathcal{AMS}$-LaTeX 宏包，请在 fontenc 和 babel 宏包之前加载。

和基于 Latin 语言的多语言文档中引入 latin 字体是很有意义的。选择不同语言的时候，babel 会切换到合适的字体编码方案。

除了断字、改变自动产生的文本字符、激活某些语言特定的排版规则（象 `\frenchspa cing` ），babel 还提供一些命令，能够按照 Bulgarian、Russian 或者 Ukrainian 语言的标准排版。

对于这三种语言，提供了语言特有的标点符号：cyrillic 书面破折号（比 latin 破折号窄一点，周围有小空格），口语破折号、引号以及帮助断字的一些命令，参看表 **??**。

表 2.6: The extra definitions made by Bulgarian, Russian, and Ukrainian options of babel

`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`"---`	Cyrillic emdash in plain text.
`"--~`	Cyrillic emdash in compound names (surnames).
`"--*`	Cyrillic emdash for denoting direct speech.
`""`	like ||, but producing no hyphen sign (for compound words with hyphen, e.g. |x-y| or some other signs as “disable/enable”).
`"~`	for a compound word mark without a breakpoint.
`"=`	for a compound word mark with a breakpoint, allowing hyphenation in the composing words.
`",`	thinspace for initials with a breakpoint in following surname.
`"‘`	for German left double quotes (looks like „).
`"’`	for German right double quotes (looks like “).
`"<`	for French left double quotes (looks like ≪).
`">`	for French right double quotes (looks like ≫).

babel 宏包的 Russian 和 Ukrainian 选项定义了 `\Asbuk` 和 `\asbuk` 命令，和 `\Alph` 及 `\alph` 有点相似，生成大写和小写的俄语和乌克兰字符（无论当前激活的语言是什么）。Bulgarian 选项提供 `\enumBul`、`\enumLat`（`\enumEng`）命令，使得 `\Alph` 和 `\alph` 命令生成保加利亚语或 Latin (English) 字符。使用 Bulgarian语言选项时，`\Alph` 和 `\alph` 缺省的方式是生成保加利亚字母的符号。

2.6 单词的间隔

LaTeX 通过调整单词间的间隔，使得输出的右边界对齐；在句子末尾插入稍大得空间，使文本更具可读性。LaTeX 假定句子以句号、问号或惊叹号结尾，如果句号紧跟一个大写字母，它就不视为句子的结尾，因为一般在有缩写地方，才出现句号紧跟大写字母的情况。

作者必须标明不符合这些假设的特殊情况。空格前的反斜线符号产生一个不能伸长的空格。波浪字符 '~' 也产生一个不能伸长的空格，并且禁止断行。如果句子末尾的句号前是一个大写字母，使用 \@ 命令说明这个句号是句子的末尾。

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?

命令

```
\frenchspacing
```

能禁止在句号后插入额外的空间，它告诉 LaTeX 在句号后不要插入比正常字母更多的空间。除了参考文献，这在非英语语言中非常普遍。如果使用了 \frenchspacing，命令 \@ 就不必要了。

2.7 标题、章和节

为便于读者理解，应该把文档划分为章，节和子节。LaTeX 用特殊的命令实现这个功能，这些命令把节的标题作为参量。我们要做的是按正确次序使用它们。

对 article 风格的文档，有下列分节命令：

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

如果想把文档分成几个部分，但是不改变原来章和节的序号，可以使

用

`\part{...}`

使用 report 或者 book cls 的时候，顶层的节命令变为了

`\chapter{...}`

article cls 没有 chapter 的概念，所以可以把 article 作为 book 中的一章。节之间的距离、序号还有标题的字体大小是 LaTeX 自动设置好的。

还有两个特殊的分节命令：

- \part 不影响章的序号。
- \appendix 不带参量，只是把章的序号用字母表示[10]

LaTeX 在文档编译的最后一个环节中，提取节的标题和页码以生成目录。命令

`\tableofcontents`

在其出现的位置插入目录。为了得到正确的目录内容，一个新文档必须编译两次。有时还要编译三次，需要时 LaTeX 会提醒你。

上面列出的分节命令也以“带星”的形式出现。“带星”的命令通过在命令名称后加 * 来实现。它们生成的节标题既不出现于目录，也不带序号。例如，命令 \section{Help} 的“带星”形式为 \section*{Help}。

目录中出现的标题，一般与输入的文本完全一致。有时这是不可能的，因为标题太长排不进目录。在这种情况下，目录的条目可由真实标题前的可选参量确定。

```
\chapter[Title for the table of contents]{A long
    and especially boring title, shown in the text}
```

[10]在 article 风格的文档中会改变节的序号。

整篇文档的 title 由

`\maketitle`

生成，title 中的所有内容必须在使用 `\maketitle` 之前定义

`\title{...}`, `\author{...}` 或者 `\date{...}`

`\author` 的参量中，可是输入几个用 `\and` 分开的名字。

上面提到的一些例子，可以在第 9 页的 1.2 图中找到。

除了上面解释的分节命令，LaTeX 2_ε 在 book 风格的文档中，还使用了其他三个命令，它们对划分文档的结构划分很有用，能按照 book 的格式改变章的标题和页码：

`\frontmatter` 是在 `\begin{document}` 后第一个使用的命令。它把页码更换为罗马数字。对于正文前的内容普遍使用带星的命令（例如，`\chapter*{Preface}`），阻止 LaTeX 对它们排序。

`\mainmatter` 出现在书的第一章前面。它打开阿拉伯页码计数器，并对页码重新计数。

`\appendix` 标志书中附录材料的开始。该命令后的各章序号改用字母标记。

`\backmatter` 出现在书中最后一部分内容的前面，如参考文献和索引。在标准文档类型中，它对页面没有什么影响。

2.8 交叉引用

在 book、report 和 article 中，需要对图、表和文本的特殊段落进行交叉引用。LaTeX 提供如下交叉引用命令

`\label{`*marker*`}`, `\ref{`*marker*`}` and `\pageref{`*marker*`}`

marker 是使用者选定的标识符。如果在 section、subsection、figure 、table 或者 theorem 后面输入 `\label` 命令，LaTeX 把 `\ref` 替换为相应的序号。`\pageref` 命令输出 `\label` 输入处的页码。[11]和章节标题一样，使用的序号是前面编译所产生。

[11] 注意这些命令并不知道它们指向什么。命令 `\label` 只是保存上一次自动产生的序号。

```
对~this~子节的引用
\label{sec:this} looks like:
``see section~\ref{sec:this} on
page~\pageref{sec:this}.''
```

对 this 子节的引用 looks like: “see section 2.8 on page 37.”

2.9 脚注

利用命令

`\footnote{`*footnote text*`}`

把脚注内容排印于当前页的页脚位置。脚注命令总是置于（put）[12]要标柱的单词或句子的后面。对一个句子或句子的一部分使用的脚注应当放到逗号或句号后面。[13]

```
Footnotes\footnote{This is
  a footnote.} are often used
by people using \LaTeX.
```

Footnotes[a] are often used by people using LaTeX.

[a] This is a footnote.

2.10 强调

如果文本是用打字机键入的，用下划线来强调重要的单词。

`\underline{`*text*`}`

但是在印刷的书中，用斜体字体排印要强调的单词。LaTeX 提供命令

`\emph{`*text*`}`

来强调文本。这些命令对其参量的实际作用效果依赖于它的上下文：

[12] “put” 是最常使用的英语单词之一。

[13] 注意，脚注会把读者的注意力从文档的正文引开。因为我们是好奇的动物，每个人都会阅读脚注。所以为什么不把你想说的所有东西都写入正文中？ [14]

[14] 路标不必走向它指向的地方 :-).

```
\emph{If you use
  emphasizing inside a piece
  of emphasized text, then
  \LaTeX{} uses the
  \emph{normal} font for
  emphasizing.}
```

If you use emphasizing inside a piece of emphasized text, then LaTeX uses the normal *font for emphasizing.*

请注意 LaTeX 中强调和使用不同字体的不同效果:

```
\textit{You can also
  \emph{emphasize} text if
  it is set in italics,}
\textsf{in a
  \emph{sans-serif} font,}
\texttt{or in
  \emph{typewriter} style.}
```

You can also emphasize *text if it is set in italics,* in a *sans-serif* font, or in *typewriter* style.

2.11 Environments

`\begin{`*environment*`}` *text* `\end{`*environment*`}`

其中，*environment* 是环境的名称。只要保证嵌套次序，Environment 可以多次嵌套。

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

在下面几节中，对所有重要的环境都做了解释。

2.11.1 Itemize, Enumerate, and Description

`itemize` 环境用于简单的列表，`enumerate` 环境用于带序号的列表，`description` 环境用于带描述的列表。

```
\flushleft
\begin{enumerate}
\item You can mix the list
environments to your taste:
\begin{itemize}
\item But it might start to
look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not
become smart because they are
in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
\end{enumerate}
```

1. You can mix the list environments to your taste:
   - But it might start to look silly.
   - With a dash.
2. Therefore remember:

   Stupid things will not become smart because they are in a list.

   Smart things, though, can be presented beautifully in a list.

2.11.2 Flushleft, Flushright, and Center

`flushleft` 和 `flushright` 环境分别产生靠左排列和靠右排列的段落。`center` 环境产生居中的文本。如果你不输入命令 `\\` 指定断行点，LaTeX 将自行决定。

```
\begin{flushleft}
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushleft}
```

This text is
left-aligned. LaTeX is not trying to make each line the same length.

```
\begin{flushright}
This text is right-\\aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushright}
```

This text is right-
aligned. LaTeX is not trying to make each line the same length.

```
\begin{center}
At the centre\\of the earth
\end{center}
```

At the centre
of the earth

2.11.3 Quote, Quotation, and Verse

quote 环境对引述、重要断语和例子的很重要。

```
A typographical rule of thumb
for the line length is:
\begin{quote}
On average, no line should
be longer than 66 characters.
\end{quote}
This is why \LaTeX{} pages have
such large borders by default and
also why multicolumn print is
used in newspapers.
```

A typographical rule of thumb for the line length is:

> On average, no line should be longer than 66 characters.

This is why LaTeX pages have such large borders by default and also why multicolumn print is used in newspapers.

有两个类似的环境：quotation 和 verse 环境。quotation 环境用于超过几段的较长引用，因为它对段落进行缩进。在诗歌中断行很重要,verse 环境比较适合诗歌。在每一行的末尾用 \\ 断行，每一段后留一空行。

```
I know only one English poem by
heart. It is about Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty had a great fall.\\
All the King's horses and all
the King's men\\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

I know only one English poem by heart. It is about Humpty Dumpty.

> Humpty Dumpty sat on a wall:
> Humpty Dumpty had a great fall.
> All the King's horses and all the King's men
> Couldn't put Humpty together again.

2.11.4 逐字打印

位于 \begin{verbatim} 和 \end{verbatim} 之间的文本将直接打印，包括所有的断行和空白，就像在打字机上键入一样，不执行任何 LaTeX 命令。

在一个段落中，类似的功能可由

\verb+*text*+

完成。+ 仅是分隔符的一个例子。除了 * 和空格，可以使用任意一个字符。

这本册子中的许多例子是用这个命令排印的。

```
The \verb|\ldots| command \ldots

\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

The \ldots command ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
the starred version of
the      verbatim
environment emphasizes
the spaces   in the text
\end{verbatim*}
```

```
the␣starred␣version␣of
the␣␣␣␣␣␣␣verbatim
environment␣emphasizes
the␣spaces␣␣␣in␣the␣text
```

带星的命令 \verb 能以类似的方式使用：

```
\verb*|like   this :-) |
```

```
like␣␣␣this␣:-)␣
```

verbatim 环境和 \verb 命令不能在其他命令的参数中使用。

2.11.5 表格

tabular 环境能用来排印带有水平和铅直表格线的漂亮表格。LaTeX 自动确定每一列的宽度。

命令

\begin{tabular}[*pos*]{*table spec*}

的参量 *table spec* 定义了表格的式样。用 l 产生左对齐的列，用 r 产生右对齐的列，用 c 产生居中的列；用 p{emphwidth} 产生相应宽度、包含自动断行文本的列；| 产生铅直表线。

一列中的文本太宽一页放不下的话，LaTeX 不会自动换行的。用 p{*width*} 定义一个特殊的列，可以象段落中一样自动换行。

pos 参量标明了表格中文字基线和表格线的相对垂直位置。可以使用 t，b 或 c 来分别表示顶对齐，底对齐和中心对齐。

在 tabular 环境中，用 texttt& 跳入下一列，用 \\ 开始新的一行，用 \hline 插入水平表线。用 \cline{*j*- *i*} 可添加部分表线，其中 j 和 i 分别表示表线的起始列和终止列的序号。

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Welcome to Boxy's paragraph.
We sincerely hope you'll
all enjoy the show.\\
\hline
\end{tabular}
```

Welcome to Boxy's paragraph. We sincerely hope you'll all enjoy the show.

表格列分隔符可由 `@{...}` 构造。这个命令去掉表列之间的间隔，代之为两个花括号间的任何输入。通常是用来处理下面要解释的十进制数对齐问题，还可以应用在用 `@{}` 压缩表列两端空间。

```
\begin{tabular}{@{} l @{}}
\hline
no leading space\\
\hline
\end{tabular}
```

no leading space

```
\begin{tabular}{l}
\hline
leading space left and right\\
\hline
\end{tabular}
```

leading space left and right

由于没有内建的方法使十进制数按小数点对齐，[15]我们可以使用两列“作弊”达到这个目的：整数部分右对齐，小数左对齐。\begin{tabular}

[15]如果系统安装了 'tools' 包，请看一下宏包 dcolumn。

行中的命令 `@{.}` 使得“.”取代了表列间的正常间隔，从而给出了按小数点对齐的效果。不要忘记用表列分隔符 (`&`) 取代十进制数的小数点！使用命令 `\multicolumn` 可在数值“表列”上放置一个表列标签。

```
\begin{tabular}{c r @{.} l}
Pi expression       &
\multicolumn{2}{c}{Value} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Pi expression	Value
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

用表格环境排版的内容总是待在同一页上。如果要排印一个长表格，可以看一下 supertabular 和 longtabular 环境。

2.12 浮动体

现在大多数出版物含有许多图片和表格。由于不能把它们分割在不同的页面上，所以需要专门的处理。如果一个图片或一个表格太大在当前页面排不下，一个解决办法就是每次新开一页。这个方法会在页面上留下部分空白，效果看起来很差。

对于在当前排不下的任何一个图片或表格，可以把它们“浮动”到下一页，与此同时当前页面用正文文本填充。LaTeX 提供了两个浮动体环境；一个用于图片，一个用于表格。要充分发挥这两个环境的优越性，应该大致了解 LaTeX 处理浮动体的内在原理。但是 LaTeX 总是不能把浮动体放在你想要的地方，因此浮动有可能成为令人沮丧的一个部分。

首先看一下 LaTeX 提供的浮动命令：

环境 figure 或 table 中的任何内容都将被视为浮动内容。两个浮动环境都支持可选参数

\begin{figure}[*placement specifier*] or \begin{table}[*placement specifier*]

称为位置说明符，告诉 LaTeX 浮动体应该放置到哪些地方，它由浮动允许放置参量中的字符组合而成。见表 2.7。

一个表格可以由如下命令

```
\begin{table}[!hbp]
```

开始，放置说明符 [!hbp] 允许 LaTeX 把表格放在命令出现的位置 (h)，放在某页的底部 (b)，或放在一个专门的浮动页上 (p)，如果效果不好甚至可以使用全部参数 (!)。如果没有给定放置说明符，缺省值为 [tbp]。

LaTeX 将按照作者提供的放置说明符，安排它遇到的每一个浮动体。如果浮动体不能放在当前页，就把它寄存在图片或表格序列中[16]。当新的一页开始的时候，LaTeX 首先检查是否可能用等待序列中的浮动体填充一个专门的“浮动”页面。如果不可能，就像对待刚在文本中出现的浮动体一样，处理等待序列中的第一个浮动体：LaTeX 重新尝试按照其相应的放置说明符（除了不再可能的‘h’）来处理它。文本中出现的任何一个新浮动体寄存在相应的等待序列中。对于每一种浮动体，LaTeX 保持它们出现的顺序。这就是为什么一个不能安排的图片会把所有后来的图片都推到文档末尾的原因。所以：

[16] “先进先出”序列！

表 2.7: 浮动允许放置参量

说明符	浮动体允许放置位置...
h	浮动体就放在当前位置。这主要用于小浮动体。
t	放在页面顶部
b	放在页面底部
p	放在一专门页面，仅含浮动体。
!	忽略阻止浮动体放置的大多数内部参数[a]。

注意：pt 和 em 都是 TeX 长度单位。第 121 页的表 6.5 含有更多的有关信息。

[a] 例如一页中允许的浮动体的最大个数。

> 如果 LaTeX 没有像你期望的那样安排浮动体，经常是有一个浮动体堵塞了两个等待序列中的某一个。

仅给定单个放置说明符是可以的，但可能会引发问题。如果在指定的位置安排不了，它就会成为障碍，堵住后续的浮动体。特别是不要单独使用参量 [h]，在 LaTeX 最近的版本中，它的效果很差，已被 [ht] 自动替换。

虽然对浮动体问题已经作了些说明，对 `table` 和 `figure` 环境还有些内容要交代。利用

`\caption{`*caption text*`}`

命令，可以给浮动体加一个标题，LaTeX 会自动加上序号和字符“图”或“表”。

`\listoffigures` and `\listoftables`

两个命令用起来和 `\tableofcontents` 命令类似，分别排印图目录和表目录。这些目录使用完整的标题，如果使用长标题，就必须再用一个能放进目录的，较短的标题。即在 `\caption` 命令后面的括号内输入较短的标题。

```
\caption[Short]{LLLLLooooooonnnnnggggggg}
```

利用 `\label` 和 `\ref`，在文本中可以为浮动体创建交叉引用。

下面的例子画一个方形，并将它插入文档。如果想在文档中为你完成以后打算粘上的图片保留空间，你可以利用这个例子。

```
Figure~\ref{white} is an example of Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Five by Five in Centimetres.} \label{white}
\end{figure}
```

在上面的例子中，为了把图片放在当前位置 (`h`) [17]，LaTeX 会做各种努力 (`!`) 如果不行，就试图将图片安排在页面的底部 (`b`)。如果不能将图片安排在当前页面，它将决定是否可能开一个浮动页面以安排这张图片，可能还有来自表格等待序列中的一些表格。如果没有足够的材料填充一个专门浮动页

[17] 假设图片等待序列已空。

面，LaTeX 就开一个新页，像对文本中刚出现的图片一样，再一次处理这个图片。

在某些情况下，可能需要使用命令

`\clearpage` or even the `\cleardoublepage`

使 LaTeX 立即安排等待序列中所有剩下的浮动体，并且开一新页。`\cleardoublepage` 会命令 LaTeX 新开 right-hand 页面。

在本书的后面，将介绍如何在 LaTeX 2_ε 文档中插入 PostScript 图形。

2.13 保护脆弱命令

作为 `\caption` 或 `\section`）等命令的参量的文本，可能在文档中出现多次（例如，在文档的目录和正文中）。一些命令会在用于类似于 `\section` 的命令时失效，被称为脆弱命令，象`\footnote` 或 `\phantom` 都是脆弱命令。这些脆弱命令需要被保护 (难道我们不需要吗？)。把 `\protect` 命令放在前面，就可以起到保护作用。

`\protect` 仅仅保护紧跟其右侧的命令，连它的参量也不惠及。不过大多数情形下，可以重复使用 `\protect`。

```
\section{I am considerate
      \protect\footnote{and protect my footnotes}}
```

第三章

数学公式排版

准备好了！在这一章里，我们将要领略 TEX 的强项：数学排版。但是请注意，这里讨论的只是一些皮毛，然而，这里解释的内容对于许多人来说都是足够的。如果你在这里无法找到你的数学排版问题的答案，请不要沮丧，极有可能你的问题在 $\mathcal{AMS}$-LATEX[1] 中涉及到了。

3.1 总论

LATEX 使用一种特殊的模式来排版数学符号和公式。它们可以放在段落中，或者拆开段落来排版。段落中的数学文字应该放在 `\(` 和 `\)`， `$` 和 `$`，或者 `\begin{math}` 和 `\end{math}` 之间。

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
$c^{2}=a^{2}+b^{2}$
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $c^2 = a^2 + b^2$

```
\TeX{} is pronounced as
\(\tau\epsilon\chi\).\\[6pt]
100~m$^{3}$ of water\\[6pt]
This comes from my
\begin{math}\heartsuit\end{math}
```

TEX is pronounced as $\tau\epsilon\chi$.

100 m^3 of water

This comes from my $\heartsuit$

[1] 美国数学学会编写的强大的 LATEX 扩展。本章里很多示例都是利用的这个扩展。它被包含在所有最近的 TEX 发行中。如果你找不到它，请去 `macros/latex/required/amslatex` 下载。

如果想让较大的数学方程或公式与段落的其余部分分开，那么就可以使用显示式样来排版，而不是将段落拆开。为此，可以把公式放在 \[和 \]，或者 \begin{displaymath} 和 \end{displaymath} 之间。

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
or you can type less with:
\[a+b=c\]
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach:

$$c^2 = a^2 + b^2$$

or you can type less with:

$$a + b = c$$

如果希望 LaTeX 为公式进行编号，就要用 equation 环境。你还可以\标记（label）一个公式编号并在其它地方利用 \ref 或者 \eqref 来引用该公式。

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
From (\ref{eq:eps}), we gather
\ldots{}From \eqref{eq:eps} we
do the same.
```

$$\epsilon > 0 \tag{3.1}$$

From (3.1), we gather ... From (3.1) we do the same.

请注意排版（typeset）的公式和显示（displayed）的公式之间的区别:

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

There are differences between *math mode* and *text mode*. For example, in *math mode*:

1. Most spaces and line breaks do not have any significance, as all spaces are either derived logically from the mathematical expressions, or have to be specified with special commands such as \,, \quad or \qquad.

2. Empty lines are not allowed. Only one paragraph per formula.

3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using the `\textrm{...}` commands (see also section 3.7 on page 58).

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \qquad x^2 \geq 0 \tag{3.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{for all }x\in\mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbf{R} \tag{3.3}$$

Mathematicians can be very fussy about which symbols are used: it would be conventional here to use 'blackboard bold', which is obtained using `\mathbb` from the package amsfonts or amssymb. The last example becomes

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{for all }x\in\mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbb{R}$$

3.2 Grouping in Math Mode

Most math mode commands act only on the next character, so if you want a command to affect several characters, you have to group them together using curly braces: `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \tag{3.4}$$

3.3 Building Blocks of a Mathematical Formula

This section describes the most important commands used in mathematical typesetting. Take a look at section 3.10 on page 61 for a detailed list of commands for typesetting mathematical symbols.

Lowercase Greek letters are entered as `\alpha`, `\beta`, `\gamma`, ..., uppercase letters are entered as `\Gamma`, `\Delta`, ...[2]

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Exponents and Subscripts can be specified using the `^` and the `_` character.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq {e^x}^2$
```

$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a^3_{ij}$
$e^{x^2} \neq {e^x}^2$

The **square root** is entered as `\sqrt`; the n^{th} root is generated with `\sqrt[`n`]`. The size of the root sign is determined automatically by LaTeX. If just the sign is needed, use `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x} \qquad \sqrt{x^2+\sqrt{y}} \qquad \sqrt[3]{2}$
$\surd[x^2 + y^2]$

The commands `\overline` and `\underline` create **horizontal lines** directly over or under an expression.

```
$\overline{m+n}$
```

$\overline{m+n}$

The commands `\overbrace` and `\underbrace` create long **horizontal braces** over or under an expression.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$\underbrace{a+b+\cdots+z}_{26}$

[2] There is no uppercase Alpha defined in LaTeX 2_ε because it looks the same as a normal roman A. Once the new math coding is done, things will change.

To add mathematical accents such as small arrows or tilde signs to variables, you can use the commands given in Table 3.1 on page 61. Wide hats and tildes covering several characters are generated with `\widetilde` and `\widehat`. The `'` symbol gives a prime.

```
\begin{displaymath}
y=x^{2}\qquad y'=2x\qquad y''=2
\end{displaymath}
```

$$y = x^2 \qquad y' = 2x \qquad y'' = 2$$

Vectors often are specified by adding small arrow symbols on top of a variable. This is done with the `\vec` command. The two commands `\overrightarrow` and `\overleftarrow` are useful to denote the vector from A to B.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec{a} \quad \overrightarrow{AB}$$

Usually you don't typeset an explicit dot sign to indicate the multiplication operation; however sometimes it is written to help the reader's eyes in grouping a formula. You should use `\cdot` in these cases:

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2
    {\tau}_1 \cdot {\tau}_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

Names of log-like functions are often typeset in an upright font, and not in italics as variables are, so LaTeX supplies the following commands to typeset the most important function names:

```
\arccos  \cos   \csc  \exp  \ker    \limsup  \min
\arcsin  \cosh  \deg  \gcd  \lg     \ln       \Pr
\arctan  \cot   \det  \hom  \lim    \log      \sec
\arg     \coth  \dim  \inf  \liminf \max      \sin
\sinh    \sup   \tan  \tanh
```

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

For the modulo function, there are two commands: `\bmod` for the binary operator "a mod b" and `\pmod` for expressions such as "$x \equiv a \pmod{b}$."

```
$a\bmod b$\\
$x\equiv a \pmod{b}$
```

$a \bmod b$
$x \equiv a \pmod{b}$

A built-up **fraction** is typeset with the `\frac{...}{...}` command. Often the slashed form 1/2 is preferable, because it looks better for small amounts of 'fraction material.'

```
$1\frac{1}{2}$~hours
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

$1\frac{1}{2}$ hours

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

To typeset binomial coefficients or similar structures, you can use the command `\binom` from the amsmath package.

```
\begin{displaymath}
\binom{n}{k}\qquad\mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k} \qquad \mathrm{C}_n^k$$

For binary relations it may be useful to stack symbols over each other. `\stackrel` puts the symbol given in the first argument in superscript-like size over the second, which is set in its usual position.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

The **integral operator** is generated with `\int`, the **sum operator** with `\sum`, and the **product operator** with `\prod`. The upper and lower limits are specified with `^` and `_` like subscripts and superscripts. [3]

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{displaymath}
```

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \prod_{\epsilon}$$

[3] $\mathcal{AMS}$-LaTeX in addition has multi-line super-/subscripts

To get more control over the placement of indices in complex expressions, amsmath provides two additional tools: the `\substack` command and the `subarray` environment:

```
\begin{displaymath}
\sum_{\substack{0<i<n \\ 1<j<m}}
   P(i,j) =
\sum_{\begin{subarray}{l} i\in I\\
         1<j<m
      \end{subarray}}     Q(i,j)
\end{displaymath}
```

$$\sum_{\substack{0<i<n \\ 1<j<m}} P(i,j) = \sum_{\substack{i\in I \\ 1<j<m}} Q(i,j)$$

TEX provides all sorts of symbols for **braces** and other delimiters (e.g. $[\ \langle\ \|\ \updownarrow$). Round and square braces can be entered with the corresponding keys and curly braces with `\{`, but all other delimiters are generated with special commands (e.g. `\updownarrow`). For a list of all delimiters available, check Table 3.8 on page 65.

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$$a,b,c \neq \{a,b,c\}$$

If you put the command `\left` in front of an opening delimiter or `\right` in front of a closing delimiter, TEX will automatically determine the correct size of the delimiter. Note that you must close every `\left` with a corresponding `\right`, and that the size is determined correctly only if both are typeset on the same line. If you don't want anything on the right, use the invisible '`\right.`'!

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{ 1-x^{2} } \right) ^3$$

In some cases it is necessary to specify the correct size of a mathematical delimiter by hand, which can be done using the commands `\big`, `\Big`, `\bigg` and `\Bigg` as prefixes to most delimiter commands.[4]

[4] These commands do not work as expected if a size changing command has been used, or the `11pt` or `12pt` option has been specified. Use the exscale or amsmath packages to correct this behaviour.

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$$\Big((x+1) (x-1) \Big) ^{2}$$

$$\big(\Big(\bigg(\Bigg(\quad \big\}\Big\}\bigg\}\Bigg\} \quad \big\|\Big\|\bigg\|\Bigg\|$$

There are several commands to enter **three dots** into a formula. `\ldots` typesets the dots on the baseline and `\cdots` sets them centred. Besides that, there are the commands `\vdots` for vertical and `\ddots` for diagonal dots. You can find another example in section 3.5.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

3.4 Math Spacing

If the spaces within formulae chosen by TEX are not satisfactory, they can be adjusted by inserting special spacing commands. There are some commands for small spaces: `\,` for $\frac{3}{18}$ quad (␣), `\:` for $\frac{4}{18}$ quad (␣) and `\;` for $\frac{5}{18}$ quad (␣). The escaped space character `\␣` generates a medium sized space and `\quad` (␣) and `\qquad` (␣) produce large spaces. The size of a `\quad` corresponds to the width of the character 'M' of the current font. The `\!` command produces a negative space of $-\frac{3}{18}$ quad (␣).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
instead of
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

$$\iint_{D} g(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

instead of

$$\int\int_{D} g(x,y)\mathrm{d}x\mathrm{d}y$$

Note that 'd' in the differential is conventionally set in roman.

$\mathcal{AMS}$-LATEX provides another way for fine-tuning the spacing between multiple integral signs, namely the `\iint`, `\iiint`, `\iiiint`, and `\idotsint`

commands. With the amsmath package loaded, the above example can be typeset this way:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \ud x \, \ud y
\end{displaymath}
```

$$\iint_D \mathrm{d}x\,\mathrm{d}y$$

See the electronic document testmath.tex (distributed with $\mathcal{AMS}$-LaTeX) or Chapter 8 of *The LaTeX Companion* [3] for further details.

3.5 Vertically Aligned Material

To typeset **arrays**, use the `array` environment. It works somewhat similar to the `tabular` environment. The `\\` command is used to break the lines.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

The `array` environment can also be used to typeset expressions that have one big delimiter by using a "`.`" as an invisible `\right` delimiter:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{if $d>c$}\\
 b+x & \textrm{in the morning}\\
 l & \textrm{all day long}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{if } d > c \\ b + x & \textrm{in the morning} \\ l & \textrm{all day long} \end{array} \right.$$

Just as with the `tabular` environment, you can also draw lines in the `array` environment, e.g. separating the entries of a matrix:

```
\begin{displaymath}
\left(\begin{array}{c|c}
 1 & 2 \\
\hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

For formulae running over several lines or for equation systems, you can use the environments `eqnarray`, and `eqnarray*` instead of `equation`. In `eqnarray` each line gets an equation number. The `eqnarray*` does not number anything.

The `eqnarray` and the `eqnarray*` environments work like a 3-column table of the form `{rcl}`, where the middle column can be used for the equal sign, the not-equal sign, or any other sign you see fit. The `\\` command breaks the lines.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

$$\begin{aligned} f(x) &= \cos x && (3.5)\\ f'(x) &= -\sin x && (3.6)\\ \int_0^x f(y)dy &= \sin x && (3.7) \end{aligned}$$

Notice that the space on either side of the the equal signs is rather large. It can be reduced by setting `\setlength\arraycolsep{2pt}`, as in the next example.

Long equations will not be automatically divided into neat bits. The author has to specify where to break them and how much to indent. The following two methods are the most common ways to achieve this.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!}-{}
                      \nonumber\\
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$\begin{aligned} \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ - \frac{x^7}{7!} + \cdots \end{aligned} \qquad (3.8)$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} +{} }
                    \nonumber\\
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \\ + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \qquad (3.9)$$

The `\nonumber` command tells LaTeX not to generate a number for this equation.

It can be difficult to get vertically aligned equations to look right with these methods; the package amsmath provides a more powerful set of alternatives. (see `align`, `flalign`, `gather`, `multline` and `split` environments).

3.6 Phantoms

We can't see phantoms, but they still occupy some space in many people's minds. LaTeX is no different. We can use this for some interesting spacing tricks.

When vertically aligning text using `^` and `_` LaTeX is sometimes just a little bit too helpful. Using the `\phantom` command you can reserve space for characters that do not show up in the final output. The easiest way to understand this is to look at the following examples.

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{versus} \qquad
{}^{12}_{6}\textrm{C}
\end{displaymath}
```

$$ {}^{12}_{6}\mathrm{C} \qquad \text{versus} \qquad {}^{12}_{6}\mathrm{C} $$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\qquad \textrm{versus} \qquad
\Gamma_{ij}^{k}
\end{displaymath}
```

$$ \Gamma_{ij}^{k} \qquad \text{versus} \qquad \Gamma_{ij}^{k} $$

3.7 Math Font Size

In math mode, TEX selects the font size according to the context. Superscripts, for example, get typeset in a smaller font. If you want to typeset part of an equation in roman, don't use the `\textrm` command, because the font size switching mechanism will not work, as `\textrm` temporarily escapes to text mode. Use `\mathrm` instead to keep the size switching mechanism active. But pay attention, `\mathrm` will only work well on short items. Spaces are still not active and accented characters do not work.[5]

```
\begin{equation}
2^{\textrm{nd}} \quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\textrm{nd}} \quad 2^{\mathrm{nd}} \tag{3.10}$$

Sometimes you still need to tell LATEX the correct font size. In math mode, this is set with the following four commands:

`\displaystyle` (123), `\textstyle` (123), `\scriptstyle` (123) and `\scriptscriptstyle` (123).

Changing styles also affects the way limits are displayed.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathop{\mathrm{corr}}(X,Y)=\frac{\displaystyle\sum_{i=1}^n(x_i-\overline x)(y_i-\overline y)}{\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline x)^2\sum_{i=1}^n(y_i-\overline y)^2\biggr]^{1/2}}$$

This is one of those examples in which we need larger brackets than the standard `\left[ \right]` provides.

[5] The $\mathcal{AMS}$-LATEX (amsmath) package makes the `\textrm` command work with size changing.

3.8 Theorems, Laws, ...

When writing mathematical documents, you probably need a way to typeset "Lemmas", "Definitions", "Axioms" and similar structures. LaTeX supports this with the command

`\newtheorem{`*name*`}[`*counter*`]{`*text*`}[`*section*`]`

The *name* argument is a short keyword used to identify the "theorem." With the *text* argument you define the actual name of the "theorem," which will be printed in the final document.

The arguments in square brackets are optional. They are both used to specify the numbering used on the "theorem." Use the *counter* argument to specify the *name* of a previously declared "theorem." The new "theorem" will then be numbered in the same sequence. The *section* argument allows you to specify the sectional unit within which the "theorem" should get its numbers.

After executing the `\newtheorem` command in the preamble of your document, you can use the following command within the document.

`\begin{`*name*`}[`*text*`]`
This is my interesting theorem
`\end{`*name*`}`

This should be enough theory. The following examples should remove any remaining doubt, and make it clear that the `\newtheorem` environment is way too complex to understand.

```
% definitions for the document
% preamble
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
%in the document
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law 1*

Law 3 *No, No, No*

The "Jury" theorem uses the same counter as the "Law" theorem, so it gets a number that is in sequence with the other "Laws." The argument in square brackets is used to specify a title or something similar for the theorem.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
If there are two or more
ways to do something, and
one of those ways can result
in a catastrophe, then
someone will do it.\end{mur}
```

Murphy 3.8.1 *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

The "Murphy" theorem gets a number that is linked to the number of the current section.You could also use another unit, for example chapter or subsection.

3.9 Bold Symbols

It is quite difficult to get bold symbols in LaTeX; this is probably intentional as amateur typesetters tend to overuse them. The font change command `\mathbf` gives bold letters, but these are roman (upright) whereas mathematical symbols are normally italic. There is a `\boldmath` command, but *this can only be used outside mathematics mode*. It works for symbols too.

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

Notice that the comma is bold too, which may not be what is required.

The package amsbsy (included by amsmath) as well as the bm from the tools bundle make this much easier as they include a `\boldsymbol` command.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 List of Mathematical Symbols

The following tables demonstrate all the symbols normally accessible from *math mode*.

To use the symbols listed in Tables 3.12–3.16,[6] the package amssymb must be loaded in the preamble of the document and the AMS math fonts must be installed on the system. If the AMS package and fonts are not installed on your system, have a look at `macros/latex/required/amslatex`. An even more comprehensive list of symbols can be found at `info/symbols/comprehensive`.

表 3.1: Math Mode Accents.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

表 3.2: Lowercase Greek Letters.

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

[6] These tables were derived from `symbols.tex` by David Carlisle and subsequently changed extensively as suggested by Josef Tkadlec.

表 3.3: Uppercase Greek Letters.

Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

表 3.4: Binary Relations.

You can negate the following symbols by prefixing them with a `\not` command.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$ [a]	`\sqsubset` [a]	$\sqsupset$ [a]	`\sqsupset` [a]	$\bowtie$ [a]	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a] Use the latexsym package to access this symbol

表 3.5: Binary Operators.

$+$	+	$-$	-		
$\pm$	\pm	$\mp$	\mp	$\triangleleft$	\triangleleft
$\cdot$	\cdot	$\div$	\div	$\triangleright$	\triangleright
$\times$	\times	$\setminus$	\setminus	$\star$	\star
$\cup$	\cup	$\cap$	\cap	$*$	\ast
$\sqcup$	\sqcup	$\sqcap$	\sqcap	$\circ$	\circ
$\vee$	\vee , \lor	$\wedge$	\wedge , \land	$\bullet$	\bullet
$\oplus$	\oplus	$\ominus$	\ominus	$\diamond$	\diamond
$\odot$	\odot	$\oslash$	\oslash	$\uplus$	\uplus
$\otimes$	\otimes	$\bigcirc$	\bigcirc	$\amalg$	\amalg
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown	$\dagger$	\dagger
$\lhd$	\lhd ^{a}	$\rhd$	\rhd ^{a}	$\ddagger$	\ddagger
$\unlhd$	\unlhd ^{a}	$\unrhd$	\unrhd ^{a}	$\wr$	\wr

表 3.6: BIG Operators.

$\sum$	\sum	$\bigcup$	\bigcup	$\bigvee$	\bigvee	$\bigoplus$	\bigoplus
$\prod$	\prod	$\bigcap$	\bigcap	$\bigwedge$	\bigwedge	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup			$\bigodot$	\bigodot
$\int$	\int	$\oint$	\oint			$\biguplus$	\biguplus

表 3.7: Arrows.

$\leftarrow$	\leftarrow or \gets	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\rightarrow$	\rightarrow or \to	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\iff$	\iff (bigger spaces)	$\leadsto$ [a]	\leadsto [a]

[a] Use the latexsym package to access this symbol

表 3.8: Delimiters.

(	(	)	)	$\uparrow$	\uparrow	$\Uparrow$	\Uparrow
[	[or \lbrack	]	] or \rbrack	$\downarrow$	\downarrow	$\Downarrow$	\Downarrow
{	\{ or \lbrace	}	\} or \rbrace	$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\langle$	\langle	$\rangle$	\rangle	$\vert$	| or \vert	$\Vert$	\\| or \Vert
$\lfloor$	\lfloor	$\rfloor$	\rfloor	$\lceil$	\lceil	$\rceil$	\rceil
/	/	\	\backslash		. (dual. empty)		

表 3.9: Large Delimiters.

$\lgroup$	\lgroup	$\rgroup$	\rgroup	$\lmoustache$	\lmoustache	$\rmoustache$	\rmoustache
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert		

表 3.10: Miscellaneous Symbols.

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$ [a]	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$ [a]	\Box [a]	$\Diamond$ [a]	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a] Use the latexsym package to access this symbol

表 3.11: Non-Mathematical Symbols.

These symbols can also be used in text mode.

†	\dag	§	\S	©	\copyright	®	\textregistered
‡	\ddag	¶	\P	£	\pounds	%	\%

表 3.12: AMS Delimiters.

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner
$\lvert$	\lvert	$\rvert$	\rvert	$\lVert$	\lVert	$\rVert$	\rVert

表 3.13: AMS Greek and Hebrew.

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\gimel$	\gimel	$\daleth$	\daleth

表 3.14: AMS Binary Relations.

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot or \Doteq
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg or \gggtr	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\Subset$	\Subset	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because	$\varpropto$	\varpropto
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\between$	\between
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright	$\blacktriangleleft$	\blacktriangleleft
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq	$\blacktriangleright$	\blacktriangleright

表 3.15: AMS Arrows.

⇠	\dashleftarrow	⇢	\dashrightarrow	⊸	\multimap
⇇	\leftleftarrows	⇉	\rightrightarrows	⇈	\upuparrows
⇆	\leftrightarrows	⇄	\rightleftarrows	⇊	\downdownarrows
⇐	\Lleftarrow	⇛	\Rrightarrow	↿	\upharpoonleft
↞	\twoheadleftarrow	↠	\twoheadrightarrow	↾	\upharpoonright
↢	\leftarrowtail	↣	\rightarrowtail	⇂	\downharpoonleft
⇋	\leftrightharpoons	⇌	\rightleftharpoons	⇃	\downharpoonright
↰	\Lsh	↱	\Rsh	⇝	\rightsquigarrow
↫	\looparrowleft	↬	\looparrowright	↭	\leftrightsquigarrow
↶	\curvearrowleft	↷	\curvearrowright		
↺	\circlearrowleft	↻	\circlearrowright		

表 3.16: AMS Negated Binary Relations and Arrows.

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

表 3.17: AMS Binary Operators.

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup or \doublecup	$\Cap$	\Cap or \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

表 3.18: AMS Miscellaneous.

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	Ⓢ	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\mho$	\mho		

表 3.19: Math Alphabets.

Example	Command	Required package
ABCdef	\mathrm{ABCdef}	
$ABCdef$	\mathit{ABCdef}	
$ABCdef$	\mathnormal{ABCdef}	
$\mathcal{ABC}$	\mathcal{ABC}	euscript with option mathcal
$\mathscr{ABC}$	\mathscr{ABC}	mathrsfs
$\mathfrak{ABCdef}$	\mathfrak{ABCdef}	eufrak
$\mathbb{ABC}$	\mathbb{ABC}	amsfonts or amssymb

第四章

特殊功能

当处理一较大的文档时，LaTeX 提供的诸如索引自动生成，参考文献管理等特殊功能将给您的提供很大的帮助。在参考文献 *LaTeX Manual* [1] 和 *The LaTeX Companion* [3] 中您可以找到更为详尽的关于 LaTeX 特殊功能和增强功能的描述。

4.1 插入 Encapsulated PostScript 图形

LaTeX 通过 `figure` 和 `table` 环境提供了一些基本的工具，使用它们你可以处理文档中象图像和图形这样的浮动对象。其实，在基本的 LaTeX 命令和 LaTeX 扩展宏包中，有很多方法可以用来生成实际中的图形。但是对于大多数使用者，理解它们是略显困难的，故作为一份简明手册，我们这里不会深入介绍这些内容。关于这方面的更多的资料，有兴趣的读者可以参考 *The LaTeX Companion* [3] 和 *LaTeX Manual* [1] 。

比较简单易行的插图方法是先用特殊的软件包 [1] 生成所需图形，然后将它们插入文档。当然，LaTeX 包中对插图提供多种的支持，但是在这份简介中，我们仅仅讨论关于 Encapsulated PostScript(EPS) 图形的使用，因为它是最简单也是应用最为广泛的一种格式。同时为了使用 EPS 格式的图片，你最好配备一台支持 PostScript[2] 输出的打印机。为了便于在 LaTeX 文档中插入图形，由 D. P. Carlisle 制作的 graphicx 宏包提供了一系列很方便的命令，并且该宏包是被称为 “graphics” 捆集 [3] 的大家族的一分子。

[1] 例如XFig, CorelDraw!, Freehand, Gnuplot, ...

[2] 另一种可以输出 PostScript 的是 GhostScript 软件包，你可以从 `support/ghostscript` 得到。Windows 和 OS/2 用户可能会更喜欢使用 GSview 软件包。

[3] `macros/latex/required/graphics`

现在，假设您已经拥有了一台可输出 PostScript 的打印机，而且正确安装了 graphicx 宏包，这样您可以参考下面的提示，一步一步将您的图片插入到文档中：

1. 将您用其他图形软件生成的图片以 EPS 格式导出[4]。

2. 用如下的命令在文件的导言部分将 graphicx 宏包载入

   `\usepackage[`*driver*`]{graphicx}`

   这里参数 *driver* 代表你要使用的将 dvi 文件转化为 postscript 文件的转换程序的名称，目前应用最为广泛的是 `dvips` 。此参数是必须要提供的，因为在 TEX 中并没有图形插入的标准，只有知道了 *driver* 的名称， graphicx 宏包才能选择正确的方法将图形信息插入到 `.dvi` 文件中，从而让打印机理解并将 `.eps` 文件插入。

3. 使用命令

   `\includegraphics[`*key*=*value*, ...`]{`*file*`}`

   来将由参数 *file* 代表的图形文件包含到您得文档中。可选的参数由一系列用逗号隔开的键名 *keys* 和相应值 *values* 组成。 *keys* 键名参数可以用来改变插入图形的宽度、高度及旋转角度。表 4.1 列出了一些主要的键名。

表 4.1: graphicx 宏包中使用的一些键名

`width`	调整图形到指定的宽度
`height`	调整图形到指定的高度
`angle`	顺时针将图形旋转指定的角度
`scale`	缩放图形

[4] 如果您使用的软件无法以 EPS 格式导出图片，试着安装 PostScript 打印机驱动（例如 AppleLaserWriter ），并且通过它将您的图片打印到一个文件，幸运的话，你会得到一 EPS 格式的图片文件。值得注意的是，要处理的EPS图片千万不能超过一页。多留意一下你的打印机驱动，一些打印机驱动程序可以明确地设置为输出 EPS 文件。

下面的示例代码会有助于您理解这些参数的用法:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

上述命令将 test.eps 文件插入到文档中，首先将原图形顺时针旋转 90 度，然后 将图形的最终宽度调整到标准页面宽度的一半。注意在没有指定高度的情形下，宽高比例默认为 1 。高度和宽度参数也可以用绝对长度单位来指定。详细的信息可以在第 121 页的表 6.5 ，更高级的用法参见文献 [9] 和 [13] 。

4.2 参考文献

使用 thebibliography 环境可以生成文档的参考文献列表，每一参考文献条目内容以如下命令开头：

\bibitem[*label*]{*marker*}

marker 参数是用来标志文档中对书、文章、论文的引用，需要在正文中引用的时候，由下述命令实现：

\cite{*marker*}

如果不选择使用 *label* 参数，参考文献条目将会自动编号列出。跟在命令 \begin{thebibliography} 后的参数定义了为条目编号预留空间的大小。下面的示例中，参数 {99} 负责告诉 LaTeX 用于标志参考文献的条目编

号不得超过数 99 。

```
Partl~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] has proposed that ...

参考文献

[1] H. Partl: *German TEX*, TUGboat Volume 9, Issue 1 (1988)

对于大型的项目，你也许需要使用 BibTEX 程序，在大多数的 TEX 发行版本都会包含 BibTEX 程序。它能够帮你维护一个参考文献数据库，并能根据你文当中的引用提取出相关参考文献条目。 BibTEX 对参考文献的描述是基于一种样式表的概念，能够形成可视化的表示，这样您就可以按照大量预先设计好的格式来创建你的参考文献列表。

4.3 编制索引

索引对于书籍来说是很有用的一部分，有了 LATEX 和支持宏包 makeindex[5]，你可以轻松地生成自己的索引。这里我们仅简单介绍几个基本的索引生成命令，有兴趣深入的读者可以参考文献 *The LATEX Companion* [3]。

启用 LATEX 的索引生成功能，在文档的导言部分必须载入宏包 makeidx：

```
\usepackage{makeidx}
```

同时也要将如下的命令

```
\makeindex
```

放到文档的导言部分来激活索引命令。索引内容由如下命令给定：

```
\index{key}
```

这里参数 *key* 就是索引中要出现的条目，另外您需要将 index 命令放置到您最终索引条目想要指向的位置，表 4.2 给出了一些 *key* 参数引用格式的几个例子。

当 LATEX 编译你的文档时，对每一个 \index 命令，会将适当的索引项和索引项所在页码写到一个系统指定的文件，该文件与你的 LATEX 输入文件同名，但文件扩展名为 (.idx)。然后将这个生成的 .idx 文件用 makeindex 程序处理，

```
makeindex filename
```

得到一个排好序的索引列表，存储在文件扩展名为 .ind 的同名文件

[5]若系统不支持长文件名，该宏包名字为 makeidx。

表 4.2: 索引关键词格式示意。

示例	索引项	注解
`\index{hello}`	hello, 1	普通格式的索引项
`\index{hello!Peter}`	Peter, 3	'hello'下的子项
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	定义索引项的格式
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	同上
`\index{Jenny|textbf}`	Jenny, **3**	定义页码的格式
`\index{Joe|textit}`	Joe, *5*	同上
`\index{eolienne@\'eolienne}`	éolienne, 4	对重音符的处理

中，这时，若 LaTeX 再次编译输入文件，这个排好序的列表将被 LaTeX 使用如下命令

```
\printindex
```

插入到文档中。 LaTeX 2_ε 自带的 showidx 宏包可以将所有索引项显示在文档的左边空白处，当较对文档或检查索引的时候这个功能尤其方便。

必须注意， `\index` 命令若使用不当，如下面例子所示，会影响您文档的布局。

```
My Word \index{Word}. As oposed
to Word\index{Word}. Note the
position of the full stop.
```

My Word . As oposed to Word. Note the position of the full stop.

4.4 定制页眉和页脚

由Piet van Oostrum编制的 fancyhdr 宏包 [6] 提供了一些供您定制文档页眉和页脚的简单命令。如下图中所示的，本手册的页眉和页脚的制作，就是对这个宏包的简单应用。

在您定制自己的页眉和页脚时，可能遇到的最棘手的问题是使页眉随着章节的题目变化。 LaTeX 分两步来达到这个目的，在页眉和页脚的定义处，你可以使用 `\rightmark` 命令来代表当前的节名，使用 `\leftmark` 来代表当前的章名。这两个命令的值将在编译时处理 chapter 或者 section 命令时被赋值。

[6] 可从下面地址得到`macros/latex/contrib/supported/fancyhdr`.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% with this we ensure that the chapter and section
% headings are in lowercase.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}  % delete current setting for header and footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % make space for the rule
\fancypagestyle{plain}{%
   \fancyhead{} % get rid of headers on plain pages
   \renewcommand{\headrulewidth}{0pt} % and the line
}
```

图 4.1: fancyhdr 宏包的设置示例。

如果你需要更大的灵活性，可以选择不用 \chapter 命令来给 \rightmark 和 \leftmark 赋值，但是你必须用其它命令 (\chaptermark，\sectionmark, 或 \subsectionmark)来给予\rightmark 和 \leftmark 你想要定制的内容。

如果你想修改页眉中章节的名称，只需要重新定义 \chaptermark 命令即可。 图 4.1 显示了为了实现本文所显示的页眉和页脚，需要对 fancyhdr宏包的一些设置。一般情况的处理，我们建议你参考一下脚注中提到的这个宏包的一些文档。

4.5 Verbatim 宏包

在本文的前面部分，你已经对 verbatim 环境 有所了解。本节中，我们去熟悉 verbatim *package* 宏包的的功能。实际上它是针对原有 verbatim 环境 的一些局限性的重新实现，对其本身来讲，这并不特别，但是由于在 verbatim 宏包中的实现中又增加了一些新的功能，这里还是值得一提的。verbatim 宏包提供了

\verbatiminput{*filename*}

命令，允许用户插入一个 ASCII 码的文本文件，而且该文件在文档中显示效果象在一个 verbatim 环境中一样。

由于 verbatim 宏包是 'tools' 宏包集的一部分，它是预装到您的系统中的，若要详细了解这个宏包，不妨参考 [10]。

4.6 下载和安装 LaTeX 宏包

大部分的 LaTeX 安装都会包含了一批事先装好的样式宏包。如果尚不能满足要求的话，还有很多可以在网上得到。最值得一去的网站是 CTAN (http://www.ctan.org/)。

各种宏包的源文件，例如 geometry ， hyphenat 等等，一般来说都包含两个文件：一个扩展名为 .ins，另一个扩展名为 .dtx。此外，通常会有一个 readme.txt，提供对宏包的简要说明。建议使用前应该先阅读这个文件。

无论如何，当您把宏包的源文件拷贝到您的机器上之后，还需要做两件事情使得 (a) 让您的的 TeX 系统知道这个新的宏包， (b) 生成该宏包的帮助文档。下面是完成第一步的的具体步骤：

1. 对 .ins 文件运行 LaTeX 命令。解压缩生成一个 .sty 文件。
2. 把 .sty 文件放到您的 TeX 系统能找到的地方。通常是在 .../*localtexmf*/tex/latex 子目录下（Windows 或者 OS/2 用户注意应该改变斜线为反斜线）。
3. 刷新您的 TeX 系统的文件名数据库。注意具体的命令取决于你使用的 LaTeX 系统： teTeX, fpTeX – `texhash`; web2c – `maktexlsr`; MikTeX – `initexmf -update-fndb` 或者使用图形界面。

下一步是从 .dtx 中解压缩得到帮助文档：

1. 对 .dtx 文件运行 LaTeX 命令，生成一个 .dvi 文件。注意你可能需要多次运行 LaTeX 命令，直到保证文中的交叉引用正确显示。
2. 检查一下 LaTeX 命令是否在当前路径生成了一个 .idx 文件。若没有，你就可以直接跳转去执行第 5 步了。
3. 为了生成文件索引，敲入命令：
   `makeindex -s gind.ist` *name*
   (这里的参数 *name* 表示不带扩展名的你得到的 .idx 文件的主文件名)。
4. 再次对 .dtx 文件运行 LaTeX 命令。
5. 最后一步，如果需要可以将文件进一步转成 .ps 文件或者 .pdf 文件以方便阅读。

有时候，您会发现还会生成一个 .glo(glossary) 文件，这样的话您还需要在第 4 步和第 5 步之间运行下面的命令：

`makeindex -s gglo.ist -o` *name*`.gls` *name*`.glo`

切记在执行第 5 步前最后对 .dtx 文件运行一遍 LaTeX 命令。

4.7 使用 pdfLaTeX

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF 属于一种 超文本 文件格式。所谓的 超文本，其含义就象在网页中，一些被标记为超级链接的词语可以实现跳转到文档中的别的地方，或者是另一篇文档。简而言之，你通过鼠标点击这些超级链接，跳转到链接指向的目的地。这意味着在 LaTeX 中，`\ref` 和 `\pageref` 都是超级链接的实现

形式。而且象目录、索引和其他一些类似的结构，都是由超级链接组成的集合。

今天我们看到的大多数的网页都是用 HTML *(HyperText Markup Language)* 语言 *(超文本标记语言)* 写成的，但是在处理科技文档时，这种格式有两个明显的不足：

1. 在 HTML 文件中插入数学公式目前还未得到广泛的支持。虽然针对此问题，已经有一个标准问世，但是我们用的大多数浏览器，不是不支持它，或者是缺少必要的字库支持。

2. HTML 文档虽然是可以打印的，但是打印结果因浏览器的不同而相差很大。而且从打印质量方面，和现今 LaTeX 所能实现的打印效果相比，差距远不止千里。

很多人尝试着去实现从 LaTeX 文件到 HTML 文件的翻译器，而且确实有一些做地不错，可以成功地将标准的 LaTeX 文件转化为语法合格的网页文件，但是不同程度地对 LaTeX 源文件作了一些简化处理，所以当你使用 LaTeX 的一些较为复杂的功能或外接宏包时，输出结果和你所期望的结果会大相径庭。正因为此，那些希望网上发布的文档能够保持保持非常好的排版效果的作者将目光转向了 PDF *(可移植文件格式)* 文件格式。因为 PDF 格式既能支持超级链接浏览，又能保持文档的美观，而且现代的浏览器均带有可以直接显示 PDF 文件的插件。

虽然 DVI 和 PS 文件查看器支持各种操作系统平台，但是更多的人倾向于使用能够浏览 PDF 文件的 acrobat reader 和 xpdf 。所以为了为了让更多的人方便得阅读您的文档，建议您提供文件的 PDF 版本。

4.7.1 适于网上发布的 PDF 文件

由 LaTeX 源文件生成 PDF 文件相当便捷。这归功于由 Hàn Thế Thành 开发的 pdfTeX 程序。就象平常的 TeX 文件输出 DVI 文件一样， pdfTeX 能够直接输出 PDF 文件。另外还有一个程序名为 pdfLaTeX ，可以由 LaTeX 源文件直接生成 PDF 文件。

现在大多数发行的 TeX 版本，象 teTeX, fpTeX,MikTeX, TeXLive 和 CMacTeX，都会自动安装 pdfTeX 和 pdfLaTeX 这两个程序。

其实，要想在编译后生成 PDF 文件，而不是 DVI 文件，只需要将 `latex file.tex` 命令替换为 `pdflatex file.tex` 。在一些并不使用命令行

方式调用 LaTeX 的系统中，您会在 TeXControlCenter 中找到一个特殊的按钮，用来实现 pdflatex 的编译。

在 LaTeX 中，您可以通过设定文件类型的可选参数如 `a4paper` 和 `letterpaper` 来定义纸张的大小，在 pdfLaTeX 中，也可以使用这种设定方法，但是 pdfLaTeX 除了需要用于文档布局的区域大小，还需要知道纸张的实际大小。 如果你使用了 hyperref 宏包(详见 83 页)，它会自动调整页面的大小。否则，你必须在文档的导言部分手动添加如下的语句：

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

下面几节将会详细讨论 LaTeX 和 pdfLaTeX 的不同之处。概括起来，主要的不同包括三个方面：字体的使用，插入图像的格式和超级链接的手动配置。

4.7.2 字体

除了可以处理主要的 LaTeX 字体格式外， pdfLaTeX 还能够处理其他各种字体 (PK bitmaps, TrueType,PostScript type 1...) 。鉴于 bitmap PK 字体在 Acrobat Reader 中显示效果较差，建议使用 PostScript Type 1 字体，这样会产生令人满意的显示。

PostScript Type 1 的 Computer Modern 和 AMSFonts 的实现是由 Blue Sky Research and Y&Y, Inc. 开发的，后来他们将版权转给了美国数学协会。1997年初这些字体被公开，现在随绝大多数的 TeX 系统一起发布。

然而，如果您使用 LaTeX 编辑非英语文档，你就需要使用 EC, LH, 和 CB 字体 (可参见 27) 页的关于 `OT1` 字体的讨论) 。 Vladimir Volovich 开发的 cm-super 字体集包含了 EC/TC, EC Concrete, EC Bright 和 LH 等字体集，你可以从 `CTAN:/fonts/ps-type1/cm-super` 处获得。一般在 TeXLive7 和 MikTeX 中都包含它。类似地，由 Apostolos Syropoulos 开发的 type 1 CB Greek 字体，可以在 `CTAN:/tex-archive/fonts/greek/cb` 下载到。但是遗憾的是这两个字体集不能保持与 Blue Sky/Y&Y 公司的 Type1 CM 一样的打印质量。在屏幕上看起来，您的文档和使用 Blue Sky/Y&Y type 1 CM 字体一样整洁有序，通过高分辨率输出设备得到的效果仅等同于原来的 bitmap EC/LH/CB 字体。

如果您创建文档使用的语言是基于拉丁语的，那么您还有别的选择。

- 您可以选择使用 aeguill 宏包，其名字是是 *Almost European Computer Modern with Guillemets* 的缩写。只需将 `\usepackage{aeguill}` 这行

命令放到您文档的导言部分，就可以使用 AE 虚拟字体来代替 EC 字体。

- 另外您可以使用 mltex 宏包，但是要求您的 pdfTEX 是增加了 mltex 选项编译得到的。

AE 虚拟字体集，有点象 MlTEX 系统，通过创建 CM 字体中的缺失字母，并将它们按照 EC 顺序重新安排，让 TEX 系统在显示时认为它拥有全部的256字母字体集。这就保证了在大多数的系统中，用户都可以使用效果出众的 type 1 格式的 CM 字体。就象如今的 T1 编码连字符在基于拉丁的欧洲语言表示中表现不凡一样。这种方法唯一的不足之处在与人工的 AE 字符不能支持 Acrobat Reader 的 查找 功能，所以在最后生成的 PDF 文件中您不能实现带重音字母单词的查找功能。

对于俄语，一个类似的使用 C1 虚拟字体的解决方案可以在 ftp://ftp.vsu.ru/pub/tex/fo 下载。这些字体集综合了同样可以在 CTAN 网站找到的来自 Bluesky 集合中的标准 CM type 1 字体和来自 Paradissa 和 BaKoMa 集合中的 CMCYR type 1 字体。因为 Paradissa 字体仅包含俄文字母，故 C1 字体正在舍弃俄语中的象形文字。

另外一种解决方案是设法切换到其它的 PostScript type 1 字体。事实上几乎每一版本的 Acrobat Reader 都包含了几种 PostScript type 1 字体。因为这些字体有不同的字号，则文档中的文本布局自然会有些变化，而且一般来说要比空间占用效率很高的 CM 字体占用更多的空间。同时文档总的视觉一致性变差，原因是象 Times, Helvetica 和 Courier 字体集（替换的首选字体）还没有设计成象在 Computer Modern 字体集在单文档中表现的那样和谐一致。

基于以上目的，由两个现成的字体既可以使用。一个是以 *Palatino* 为其文档主体字体的 pxfonts，另一个是基于 *Times* 字体的 txfonts 包。要使用它们，只需要将如下几行语句加入到文档的导言部分即可：

```
\usepackage[T1]{fontenc}
\usepackage{pxfonts}
```

注意: 当你编译你的输入文件时，你会在 .log 文件中发现如下行所示的警告信息

```
Warning: pdftex (file eurmo10): Font eurmo10 at ... not found
```

这意味着您文中使用的字体不能被正确找到，你必须设法修正这个问题，否则生成的 PDF 文件将 不会显示带有缺失字符的页 。

字体问题，尤其是缺少质量和 type 1 格式的 CM 字体一样的 EC 字体，一直是很多人思考并致力解决的问题，相信随时都会有新的解决方案涌现。

4.7.3 使用图形

使用 graphicx 宏包来为文档插入图形，是一个明智的选择（见 71 页）。同样，只需将 *driver* 选项的名称设置为 `pdftex`，这个宏包就能在 pdfLaTeX 中大显身手了：

```
\usepackage[pdftex]{color,graphicx}
```

上面的例子中，我们启用了颜色选项，因为在网站上发布的文章中显示颜色是十分常见的。

不要过早陶醉于接踵而至的好消息，一个较坏的事实是 Encapsulated PostScript 格式的图形文件在 PdfLaTeX 中无法工作。如果你不在 `\includegraphics` 命令中额外指定扩展文件格式， graphicx 宏包将会自动查找适合自己格式的文件，当然这依赖于设置中的 *driver* 选项。对 `pdftex` 来说，这些格式包括 `.png`, `.pdf`, `.jpg`, `.mps`（ MetaPost），和 `.tif`，但是 没有 `.eps` 格式。

解决这个问题的一个简单方法是将你的 EPS 文件转化为 PDF 格式，许多系统都会提供 `epstopdf` 这一工具，用以实现这一功能。对于矢量图（画出的图）这是个不错的方案。可是对于位图（照片或扫描图片），这种方法并不高效。因为 PDF 格式本身就支持插入 PNG 和 JPEG 格式的图像。 PNG 格式对于象屏幕抓图和其他一些颜色使用较少的图像效果很好，而 JPEG 格式更适合相片，是一种占用空间很省的格式。

这里，并不建议用作图工具去画几何图形，不如用一些特定的命令语言来描绘并生成图形，象使用 MetaPost 语言。它包含在大多数的 TeX 发行中，而且有随之附带的扩展手册供您学习参考。

4.7.4 超文本链接

hyperref 宏包的可将您文档中的内部引用转化为超级链接，当然要实现它，还需要一些技巧。首先必须将 `\usepackage[pdftex]{hyperref}` 作为您的导言中的最后一个命令加入到文档的开头部分。

hyperref 宏包提供了很多的可选项，方便您定制该宏包的功能：

- 或者如下将它们放到命令的 pdftex 选项后面，并用逗号隔开
  \usepackage[pdftex]{hyperref}
- 或者用命令 \hypersetup{*options*} 新开一行来设置。

pdftex选项是必须提供的，其它的选项可根据您的需要来修改 hyperref 宏包的默认设置[7] 。下面选项列表中用右斜体书写的就是该宏包的默认值。

bookmarks (=true,*false*) 用于设置文档中书签的显示或隐藏

unicode (=false,*true*) 用于设置在 Acrobat 的书签中是否允许使用非基于拉丁语言的字符

pdftoolbar (=true,*false*) 用于设置是否显示 Acrobat 的工具栏

pdfmenubar (=true,*false*) 用于设置是否显示 Acrobat 的菜单栏

pdffitwindow (=true,*false*) 用于调整显示 PDF 文件的初始缩放比例

pdftitle (={text}) 用于定义在 Acrobat 窗口文件信息处显示的标题内容

pdfauthor (={text}) 设置 PDF 文件的作者

pdfnewwindow (=true,*false*) 定义链接指向的其它文档是否在新的窗口中打开

colorlinks (=false,*true*) 将链接用彩色框包围起来 (false)，或者仅将代表链接的文字带上颜色 (true)，其中这些链接的颜色可用下面的选项设置（给出值的是默认颜色）：

linkcolor (=red) 内部链接的颜色 (节，页等)

citecolor (=green) 引用链接的颜色 (参考文献)

filecolor (=magenta) 文件链接的颜色

urlcolor (=cyan) URL 连接的颜色 (邮件地址，网站地址)

如果这些缺省设置就很合乎您的要求，直接使用如下的命令：

```
\usepackage[pdftex]{hyperref}
```

[7] 有一点值得注意， hyperref 宏包不仅限于工作于 pdfTEX，它也可以用于普通的 LATEX，通过设置将 PDF 特征信息嵌入到 DVI 输出中。然后由 dvips 转入到 PS 文件中。最后在由 PS 文件转为 PDF 文件的过程中，由 Adobe Distiller 提取出来。

如果想显示书签，并用颜色来标记链接，使用下面的命令可以实现（等于 =true 得值可写可不写）：

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

若是为了打印的目的而生成 PDF 文件，用颜色来标记链接就不是一个很好的选择。因为最终的输出显示链接是灰色的，不容易辨认。这是可以选择使用如下的颜色框来标记，因为打印时它并不显示出来：

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

或者干脆将链接的颜色设置为黑色：

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
            citecolor=black,%
            filecolor=black,%
            linkcolor=black,%
            urlcolor=black,%
            pdftex}
```

当您想在 PDF 文件的 文件信息 处显示您提供的信息，只需要加入：

```
\usepackage[pdfauthor={Pierre Desproges}%
            pdftitle={Des femmes qui tombent},%
            pdftex]{hyperref}
```

除了用于交叉引用的自动超文本链接，您还可以嵌入更为直截了当的链接，只需要如下语句：

\href{*url*}{*text*}

例如下面的代码

```
The \href{http://www.ctan.org}{CTAN} website.
```

将会产生输出 “CTAN”，点击单词 “CTAN” 将会把你带到 CTAN 网站。

如果链接的目标，并不是一个 URL 地址，而是一个本地文件，您可以使用 \href 命令：

```
The complete document is \href{manual.pdf}{here}
```

这将产生文字“The complete document is here”，点击单词“here”将会打开文件 `manual.pdf`，注意这里文件名路径是相对于当前文档所在位置。

文章的作者也许希望读者能方便地发送电子邮件给自己，只需要在文档的标题页的 `\author` 命令中使用 `\href` 命令：

```
\author{Mary Oetiker $<$\href{mailto:mary@oetiker.ch}%
        {mary@oetiker.ch}$>$
```

注意，我们如上已将作者的的邮件地址加入到了链接中，它不仅会显示在链接中，也会显示在页面上。这样做是为了保证链接
`\href{mailto:mary@oetiker.ch}{Mary Oetiker}`
在 Acrobat 中正常显示。一旦付诸于打印，邮件地址就看不到了。

4.7.5 链接常见问题

有时编译文件时会出现如下警告信息：

```
! pdfTeX warning (ext4): destination with the same identifier
  (name{page.1}) has been already used, duplicate ignored
```

这是因为当前使用的一个计数器被重新赋值产生的。举个简单的例子，当使用 `book` 文档类中的 `\mainmatter` 命令时，会将书中第一章的首页的页码设置为 1，但是由于书的前言部分也会有页码 1 存在，所以到页码 1 的链接就不唯一了，故警告信息提示用户“忽略重复链接 .”

此时可以选择的处理方法是将 `plainpages=false` 插入到 hyperref 的选项中去。遗憾的是这种方法只对页码计数器起作用，更根本的解决方案是使用选项 `hypertexnames=false`，但是这样做会导致索引中的页码链接失效。

4.7.6 书签常见问题

文档中最终显示的书签可能总是不能达到你希望的样子。因为书签仅是由“纯文本”构成的，可供使用的字符无法与平常的 LaTeX 使用的字符集相比。Hyperref 通常很注意这个限制，所以经常会因为使用不支持的符号产生如下的警告：

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

解决这个问题的方法是由如下命令用一纯文字串取代引起麻烦的文本：

`\texorpdfstring{`*TEX text*`}{`*Bookmark Text*`}`

其实，引起这类问题的主要原因是想在书签中使用数学符号：

```
\section{\texorpdfstring{$E=mc^2$}%
        {E\ =\ mc\texttwosuperior}}
```

上面的命令将 `\section{$E=mc^2$}` 在书签显示处改为 "E=mc2"。

书签中的颜色变化也会遇到麻烦，下面的命令：

```
\section{\textcolor{red}{Red !}}
```

将会产生字符串 "redRed!"，因为命令 `\textcolor` 被编译器忽略，仅保留了它的参数 (red) 。

如果你使用下述命令：

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

结果会好得多。

LaTeX 与 pdfLaTeX 的兼容性

理想情况下，您的文档应该既可以用 LaTeX 编译，也可以使用 pdfLaTeX 编译。时刻当需要在您的文档中插入图片的时候会遇到一些问题。简单的解决方案是将 `\includegraphics` 命令参数中的文件名的扩展名 *系统地去掉* ，编译文件时系统会在文件所在的当前目录下自动寻找合适格式的文件，你需要做的就是准备不同格式的图片文件，因为 LaTeX 会寻找 `.eps` 格式的文件，而 pdfLaTeX 会选择插入扩展名为 `.png`, `.pdf`, `.jpg`, `.mps` 或者 `.tif` 的图片文件(按照括号中的排列顺序寻找)。

如果你想在 PDF 版本的文件中使用不同的代码，你可以添加如下的代码作为您文档的最开头几行：

```
\newif\ifPDF
\ifx\pdfoutput\undefined\PDFfalse
```

```
\else\ifnum\pdfoutput > 0\PDFtrue
     \else\PDFfalse
     \fi
\fi
```

它负责定义了一个特殊的命令。有了它，你可以简单地实现条件判断代码：

```
\ifPDF
  \usepackage[T1]{fontenc}
  \usepackage{aeguill}
  \usepackage[pdftex]{graphicx,color}
  \usepackage[pdftex]{hyperref}
\else
  \usepackage[T1]{fontenc}
  \usepackage[dvips]{graphicx}
  \usepackage[dvips]{hyperref}
\fi
```

如上例所示，在非 PDF 版本的文件中，依然添加了 hyperref 宏包，作用是使 `\href` 命令变得随处可用，避免了每一处使用这个命令你都需要一个条件语句来包装。

值得注意的是，最近发行的 TEX 版本中（例如 TEXLive ），调用 graphicx 和 color 宏包时需要涉及到的 `pdftex` 和 `dvips` 切换，可以根据 textttgraphics.cfg 和 color.cfg 两个配置文件自动完成。

4.8 使用 pdfscreen 来演示文档

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

通常在演示你的科研成果时，你会选择黑板、幻灯片，或者通过演示软件直接用笔记本电脑来展示。

pdfLATEX 用合成在一起的 pdfscreen 宏包来帮助用户制作 PDF 格式的演示文档，效果与用 *Powerpoint* 做出的幻灯片一样生动、多彩。而且由于 Acrobat Reader 可以在多种系统中使用， PDF 格式的演示文档的可移植性很强。

pdfscreen 类使用 graphicx, color 和 hyperref 宏包，但是需要将它们的选项调整为适于屏幕输出。

```
\documentclass[pdftex,12pt]{article}
%%% misc extensions %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[latin1]{inputenc}
\usepackage[english]{babel}
\usepackage[T1]{fontenc}
\usepackage{aeguill}
%%% pdfscreen %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[screen,panelleft,chocolate]{pdfscreen}
% Screen Format
\panelwidth=25mm
%%          height width
\screensize{150mm}{200mm}
%%          left right top  bottom
\marginsize{42mm}{8mm}{10mm}{10mm}
% Color or image for background
\overlayempty
\definecolor{mybg}{rgb}{1,0.9,0.7}
\backgroundcolor{mybg}
% Logo
\emblema{MyLogo}
%%% For PPower4 (post-processor) %%%%%%%%%%%%%%%%%%%%%%%
\usepackage{pause}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}
\begin{slide}
\begin{itemize}
\item Good News\dots \pause
\item Bad News
\end{itemize}
\end{slide}
\end{document}
```

图 4.2: pdfscreen 输入文件示例

生成这样类型的文档，一般是用 article 类，图 4.2 给出了输入文件的一个示例。首先，你需要载入 pdfscreen 宏包，并且正确设置它的可选参数：

screen ：设置为屏幕演示，若要产生一个可打印文档，应该设置为 print。

panelright 设置在屏幕的右侧生成一浏览面板。如果你想让它出现在左侧，则使用 panelleft ，若不想显示它，直接用 nopanel 即可。

french 法语或其他支持语言选项，能确保浏览按钮上面的文字正确显示。这个选项与 babel 宏包的选项集是独立无关的。如果 pdfscreen 不支持你使用的语言，你可以用 pdfscreen.cfg 来设置你浏览面板的按钮的文字。具体示例可以参见 pdfscreen.cfg..specimen 。

chocolate 设置浏览面板的颜色样式。咖啡色 (chocolate) 是默认颜色，其它可供选择的颜色还有 gray, orange, palegreen, bluelace 和 blue。

下一步你需要设置显示格式。因为一般演示文档将缩放至屏幕的实际尺寸来显示，所以需要设置总体的字体大小：

\panelwidth 定义了浏览面板的宽度。

\screensize{*width*}{*height*} 定义了包含浏览面板的总屏幕的宽度和高度。

\marginsize{*left*}{*right*}{*top*}{*bottom*} 定义了文档的页边空白。在上面的示例中，并未将文本置于中心位置，因为需要预留空间来将章节号写到页左边空白处。

使用下面的命令，可以将 pdfLaTeX 支持的图像插入到文档中作为背景图片:

\overlay{*image*}

或者你更喜欢一个空白背景，当然你也可以将背景色定义为自己喜欢的颜色，

\background{*color*}

最后，若你想将公司或组织的标志标放置到浏览面板中，可以使用如

下命令：

```
\emblema{logo}
```

若你觉得演示过程中，将你的观点逐一连续显示到屏幕更具有说服力，可以考虑使用 pause 宏包。它提供了 \pause 命令，将该命令放置到需要 Acrobat 显示暂停的文字处， pause 宏包是 ppower4 (P^4:*Pdf Presentation Post-Processor*) 系统的一部分，该系统提供对 pdfLaTeX 输出的 PDF 文件的后处理，使文档真正能够如你所愿动起来。命令行下的运行格式为：

```
ppower4 xy.pdf xyz.pdf
```

若要控制单页幻灯片上的内容的位置，使用环境 \begin{slide} ... \end{slide}，每一页幻灯片上的内容将会显示到该页的垂直正中。

编译图 4.2 中的示例代码，会产生如下的错误信息：

```
! pdfTeX warning (dest): name{contents} has been
  referenced but does not exist, replaced by a fixed one
```

这是因为浏览面板中有一按钮是指向目录的，由于该示例中并没有包含 \tableofcontents 命令，所以该链接的解析就失败了。

如果你想令目录表显示到您的浏览面板中，你可以在调用 pdfscreen 时使用选项 paneltoc 。但是只有当你的演示文档由很少而且很短的目录条目，才会产生令人满意的效果，所以最好能够提供包含在方括号中的短小的节标题。

本薄册只是介绍了 pdfscreen 和 PPower4 强大功能中的皮毛，欲要深入研习，可以参考随宏包一起的扩展文档。

第五章

创建数学图形

很多人用LaTeX 来对文本进行排版，但对创建图形却知之甚少。虽然在LaTeX 中主要从内容和结构的方面来考虑写作，但是有时反其道而行之又是很方便的。所以LaTeX 中提供了一种可以让我们通过输入文本描述来产生图形的方法，不过它有一定的局限性。为了克服LaTeX 本身产生图形能力的限制，人们又做出了一些扩展包。本节我们就来学习一下在LaTeX 中创建数学图形的基本知识以及这些扩展包的用法。

5.1 Overview

在LaTeX 中我们可以使用picture 环境来直接建立图形。关于picture 环境的细节可以在*LaTeX Manual* [1]中找到。一方面，使用picture 环境有相当严格的约束条件，比如线段的斜率和圆半径都只能选择有限的几个值，另一方面，picture 环境提供了功能强大的\qbezier 命令(“q”的意思是二次的)。因为圆、椭圆和悬链线这些常用的曲线都可以通过二次贝塞尔曲线来相当精确地近似，虽然这需要不少数学计算。另外，如果我们使用编程语言(比如JAVA)来在LaTeX 输入文件中自动生成\qbezier 命令集，那么picture 环境就变成一个相当强大的工具了。

直接在LaTeX 中“编程”生成图形不仅有严格的限制，而且常常是枯燥的工作，但这样做也有好处。因为通过这种方式产生的文件不仅字节数更少，而且不需要附加另外的图形文件。这就是通过“编程”生成图形的原因。

为了消除picture 环境作图的限制，人们创建了不同的扩展包，比如epic 、eepic(在*The LaTeX Companion* [3]等书中有介绍)或者pstricks。这些扩展包同时也大大增强了LaTeX 的作图能力。

上文中，前面的两个包epic 和eepic 仅仅是加强了picture 环境的作图能力，而pstricks 包则使用自己的作图环境，pspicture。pstricks 包的强大功能源于它大量使用了PostScript语言能力这样一个事实。另外，为了特定的用途，人们建立了大量的包。Xy-pic 包是其中之一，我在本章最后介绍它，其它多种这样的包在*The LaTeX Graphics Companion* [4](不要和*The LaTeX Companion* [3]搞混了)中做详细的讲解。

MetaPost也许是和LaTeX相关的图形工具中最强大的，它也是Donald E. Knuth编写的METAFONT 的一个孪生兄弟。MetaPost有和METAFONT同样强大和复杂的数学编程语言。和METAFONT 生成位图不同的是，MetaPost产生嵌入式的PostScript文件，LaTeX中可以导入这样的文件。MetaPost 的介绍可以在*A User's Manual for MetaPost* [15]或文献[17]中的指南部分找到。

有关LaTeX 和TeX 作图(和字体)策略的详尽叙述可以在*TeX Unbound* [16]中找到

5.2 picture 环境

By Urs Oswald <osurs@bluewin.ch>

5.2.1 基本命令

picture环境的建立可以使用下面两个命令中的一个

```
\begin{picture}(x,y)...\end{picture}
```

或

```
\begin{picture}(x,y)(x0,y0)...\end{picture}
```

x, y, x_0, y_0的单位是\unitlength。\unitlength可以随时(不能在picture环境中)用下面的命令设定

```
\setlength{\unitlength}{1.2cm}
```

\unitlength的默认值为1pt。第一对数(x,y)定义了在文档中为图像预留出矩形空间的大小。第二对数(x_0,y_0)是可选的，它定义了预留出的矩形坐下角的绝对坐标。

大多数的作图命令具有如下两种之一的格式

`\put(`x, y`){`$object$`}`

或

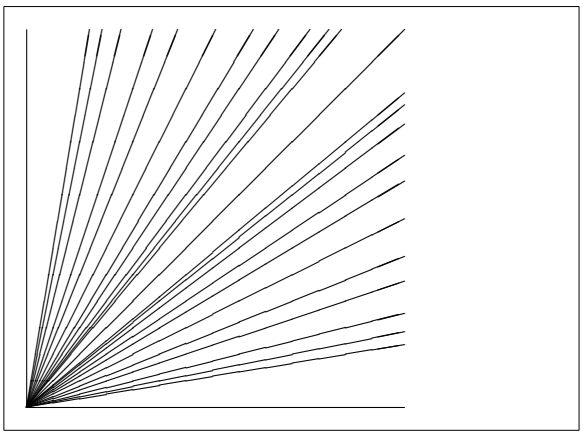
`\multiput(`x, y`)(`$\Delta x, \Delta y$`){`n`}{`$object$`}`

贝塞尔曲线是一个例外。下面的命令可以画贝塞尔曲线

`\qbezier(`x_1, y_1`)(`x_2, y_2`)(`x_3, y_3`)`

5.2.2 线段

```
\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}
```



下面的命令用来画线段

\put$(x,y)$\{\line$(x_1,y_1)$\{$length$\}\}

\line命令有两个参数：

1. 一个是方向向量,
2. 一个是线段长度。

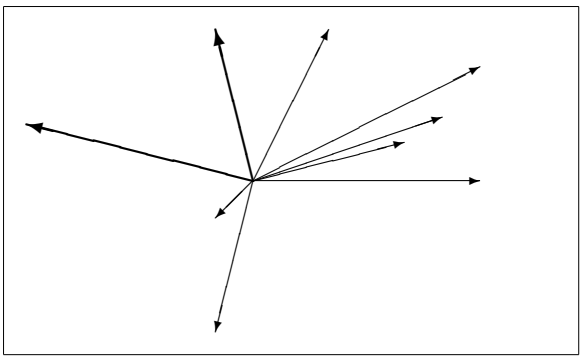
方向向量的两个分量只能取下面的整数之一

$$-6,\ -5,\ \ldots,\ 5,\ 6,$$

并且两个分量必须互质(没有除1之外的公因子)，图中显示了在第一象限中所有的25种斜率的直线。线段的长度单位是\unitlength。对于垂直的线段，长度参数就是线段的纵坐标长度，对于其它情况长度参数都为线段在横轴上的投影长度。

5.2.3 箭头

```
\setlength{\unitlength}{1mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}
```



下面的命令用来画箭头：

\put(x,y){\vector(x_1,y_1){$length$}}

对于箭头来说，它的方向向量的分量和线段相比有更加严格的限制，确切地说被限制为下面的整数：

$$-4,\ -3,\ \ldots,\ 3,\ 4.$$

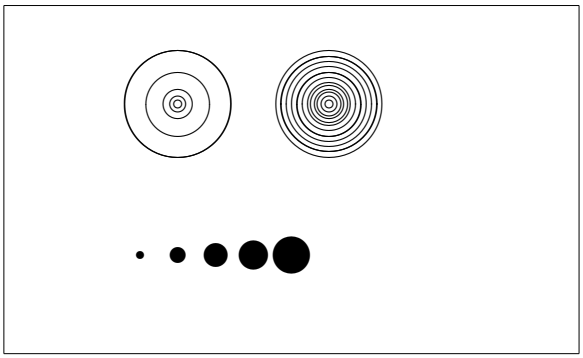
同样，两个分量也要求互质（除了1外没有公因子）。另外，请注意在上面的例子中，在指向左上角的两个箭头中使用的\thicklines命令的效果。

5.2.4 画圆

```
\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}

  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}
```



下面的命令

`\put(`x, y`){\circle{`*diameter*`}}`

以(x, y)为圆心，以*diameter*为直径（注意不是半径）画圆。在picture环境中，直径的值有一些限制，其大小最大只能是约为14 mm，并且即使小于这个值，也不是所有的值都可以的。\circle*命令会画出圆盘（实心圆）。

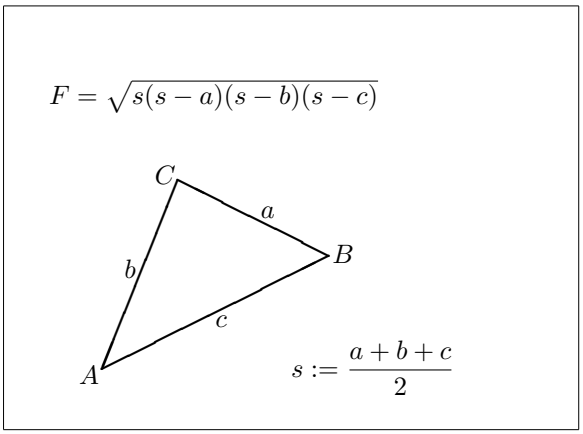
和画线段的情况一样，有时我们需要借助于另外一些扩展包，比

如eepic 或pstricks 对于这些包的详细论述，请参照 *The LaTeX Graphics Companion* [4]。

在picture环境中，还有另外一种画图的方式。如果做一些必要的计算（或者让程序去做这些），那么可以把二次Bézier曲线组合起来画出任意的圆和椭圆。在 *Graphics in LaTeX 2ε* [17] 中可以找到例子和Java原文件。

5.2.5 文本和公式

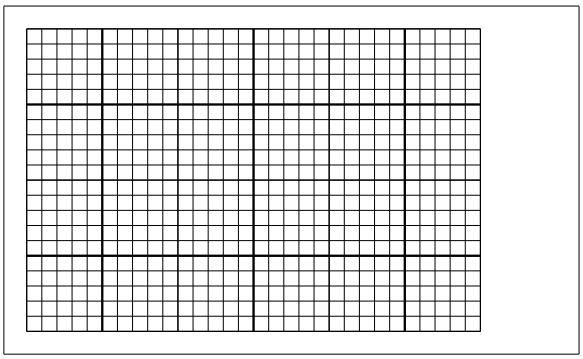
```
\setlength{\unitlength}{1cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){$A$}
  \put(4.05,1.9){$B$}
  \put(1.7,2.95){$C$}
  \put(3.1,2.5){$a$}
  \put(1.3,1.7){$b$}
  \put(2.5,1.05){$c$}
  \put(0.3,4){$F=
    \sqrt{s(s-a)(s-b)(s-c)}$}
  \put(3.5,0.4){$\displaystyle
    s:=\frac{a+b+c}{2}$}
\end{picture}
```



如例子中所示，在picture环境中，我们可以象画线或者画圆一样，用\put命令来写文本和公式。

5.2.6 \multiput和\linethickness命令

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){31}%
    {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
    {\line(1,0){30}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){7}%
    {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
    {\line(1,0){30}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){3}%
    {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
    {\line(1,0){30}}
\end{picture}
```



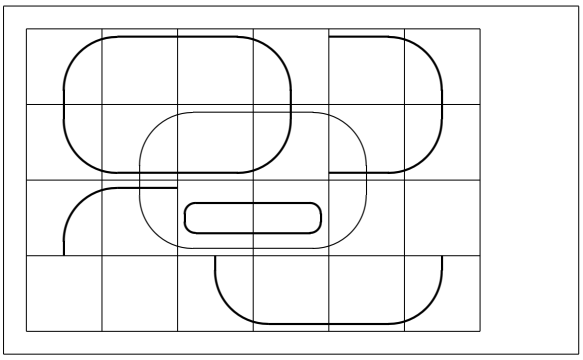
命令

`\multiput(`x, y`)(`$\Delta x, \Delta y$`){`n`}{`$object$`}`

有四个参数：开始点，相邻对象之间的移动向量，对象的数量和要画的对象。`\linethickness`命令可以用于水平和垂直的线段，但是不能用于倾斜的线段和圆。当然可以用于二次Bézier曲线。

5.2.7 卵形，\thinlines 和\thicklines 命令

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[tl]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```



命令

`\put(`x, y`){\oval(`w, h`)}`

和

`\put(`x, y`){\oval(`w, h`)[`*position*`]}`

以(x, y) 为中心画出宽为w高为h的卵形。可选的*position*参数包括`b`, `t`, `l`, `r`，分别指“上”，“下”，“左”，“右”，这些参数和组合用来画出卵形的一部分，正如例子中那样。

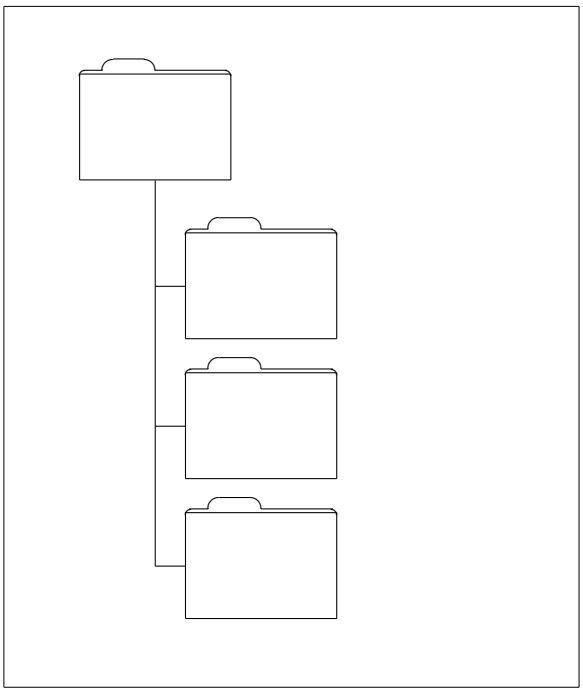
图中线的粗细可以用两类命令来控制，一个是：

`\linethickness{`*length*`}`

另一个是`\thinlines` 和`\thicklines`。但是`\linethickness{`*length*`}`命令只能用于水平和垂直直线（还有二次Bézier曲线），而`\thinlines`和`\thicklines` 命令可以用于倾斜线段、圆和卵形。

5.2.8 预定义图像块的重复使用

```
\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}% declaration
\savebox{\foldera}
  (40,32)[bl]{% definition
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}}
  \put(1,28){\oval(2,2)[tl]}
  \put(1,29){\line(1,0){5}}
  \put(9,29){\oval(6,6)[tl]}
  \put(9,32){\line(1,0){8}}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}}
  \put(39,28){\oval(2,2)[tr]}
}
\newsavebox{\folderb}% declaration
\savebox{\folderb}
  (40,32)[l]{%          definition
  \put(0,14){\line(1,0){8}}
  \put(8,0){\usebox{\foldera}}
}
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
  {\usebox{\folderb}}
\end{picture}
```



用下面的命令可以声明图像块

\newsavebox{*name*}

用下面的命令定义

\savebox{*name*}(*width,height*)[*position*]{*content*}

并且可以任意地用下面的命令画出

\put(x, y)\usebox{*name*}

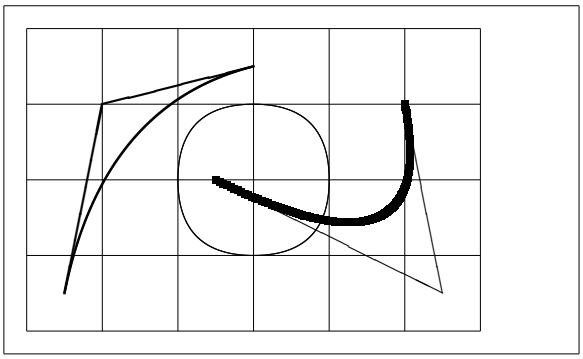
可选参数*position*定义了整个图像盒子的定位点。上面的例子中定位点参数为bl，通过这个参数，定位点就定义为图像盒子的左下角的点。其它的定义有t 为顶端的点和r 为右端的点。

参数*name* 用来指示LaTeX 中图像盒子占用的内存空间，因此名字参数就要有命令的性质，也就是像上例当中那样要包含反斜杠。另外，图像盒子可以嵌套使用，例子中，\folderb 盒子中使用了\foldera 盒子的定义。

当线段的长度小于3 mm 时，\line 命令就不能正常工作了，这时就必须使用\oval 命令。

5.2.9 二次Bézier 曲线

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}
```



上面例子中的情况说明，用4个二次Bézier曲线拼凑出一个圆的结果并不令人满意。要取得满意的结果，至少需要8个才行。同时，上图再一次演示了改变线条粗细的命令，`\linethickness` 命令用于改变水平和垂直线段的粗细，`\thinlines` 和`\thicklines` 命令用于改变倾斜线段的粗细，而这几个命令都可以用于改变二次Bézier曲线的粗细，当使用新的命令时，以前的粗细设置就被覆盖了。

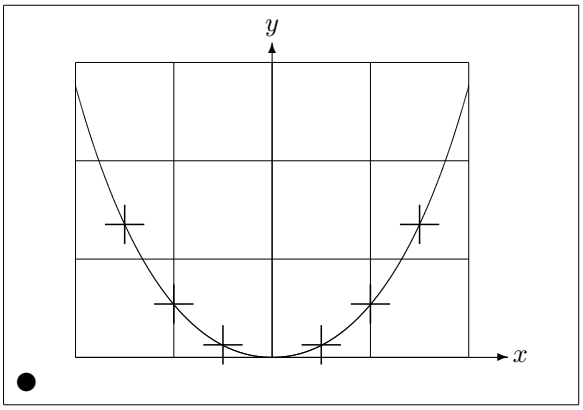
如果用$P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ 表示曲线的两个端点，m_1, m_2 表示相对的斜率，那么中间的控制点$S = (x, y)$用下面的方程可以求出：

$$\begin{cases} x &= \dfrac{m_2 x_2 - m_1 x_1 - (y_2 - y_1}{m_2 - m_1}, \\ y &= y_i + m_i(x - x_i) \qquad (i = 1, 2). \end{cases} \tag{5.1}$$

在*Graphics in LaTeX 2ε* [17]中可以找到一个Java 程序，它可以用来生成所需的`\qbezier` 命令行。

5.2.10 悬链线

```
\setlength{\unitlength}{1.3cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
  \put(-2,0){\vector(1,0){4.4}}
  \put(2.45,-.05){$x$}
  \put(0,0){\vector(0,1){3.2}}
  \put(0,3.35){\makebox(0,0){$y$}}
  \qbezier(0.0,0.0)(1.2384,0.0)
    (2.0,2.7622)
  \qbezier(0.0,0.0)(-1.2384,0.0)
    (-2.0,2.7622)
  \linethickness{.075mm}
  \multiput(-2,0)(1,0){5}
    {\line(0,1){3}}
  \multiput(-2,0)(0,1){4}
    {\line(1,0){4}}
  \linethickness{.2mm}
  \put( .3,.12763){\line(1,0){.4}}
  \put(.5,-.07237){\line(0,1){.4}}
  \put(-.7,.12763){\line(1,0){.4}}
  \put(-.5,-.07237){\line(0,1){.4}}
  \put(.8,.54308){\line(1,0){.4}}
  \put(1,.34308){\line(0,1){.4}}
  \put(-1.2,.54308){\line(1,0){.4}}
  \put(-1,.34308){\line(0,1){.4}}
  \put(1.3,1.35241){\line(1,0){.4}}
  \put(1.5,1.15241){\line(0,1){.4}}
  \put(-1.7,1.35241){\line(1,0){.4}}
  \put(-1.5,1.15241){\line(0,1){.4}}
  \put(-2.5,-0.25){\circle*{0.2}}
\end{picture}
```



在这幅图中，对称的悬链线$y = \cosh x - 1$ 的每一半都是用一个二次Bézier 曲线来模拟表示的。右半边曲线的终点为(2, 2.7622) ，终点处的斜率为$m = 3.6269$ 。再次使用方程(5.1) ，我们可以计算得到中间的控制点，结果为(1.2384, 0) 和$(-1.2384, 0)$ 。图中的小十字表示真正的悬链线上的点，通过对比发现，模拟的误差小到不超过1%，因此可以忽略不计。

本例演示了\begin{picture} 命令中可选参数的使用。图形的定义使

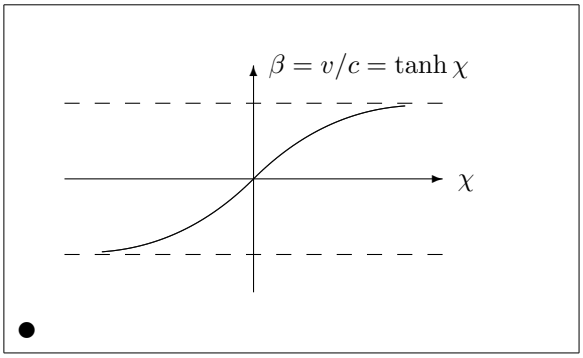
用了方便的“数学”坐标系，但是通过命令

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

定义的图形的左下角(用实心黑圆标记出来了)坐标定义为$(-2.5, -0.25)$。

5.2.11 狭义相对论中的速度

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){$\chi$}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {$\beta=v/c=\tanh\chi$}
  \qbezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qbezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}
```



上例中两条Bézier 曲线的控制点坐标是根据公式(5.1) 计算得到的。其中正向的曲线分支由参数$P_1 = (0, 0)$, $m_1 = 1$ 和$P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$ 决定。同样，上图也使用了常用的数学坐标来定义，坐下角的坐标定义为$(-3, -2)$ (用圆黑点表示)。

5.3 Xy-pic

By Alberto Manuel Brandão Simões <albie@alfarrabio.di.uminho.pt>

xy是为画图而产生的特殊的包。要使用它，只要把下面一行加到文档的导言部分就可以了：

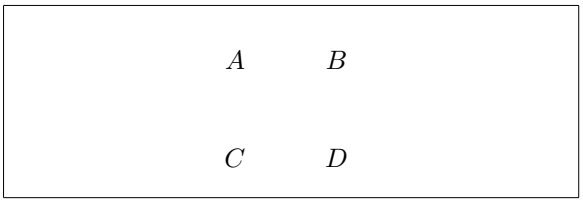
```
\usepackage[options]{xy}
```

options 是一个命令列表，包括了你要使用的Xy-pic 包中的命令。这些

参数主要在调试包的时候用到。在正常情况下，我建议使用参数all ，这样LaTeX 会载入XY 包中所有的命令。

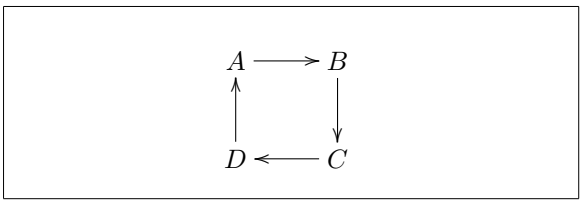
用XY-pic 画图是在基于矩阵的图板上进行的，图板上的空位以矩阵方式排列，图元素只能放置在这些空位上。

```
\begin{displaymath}
\xymatrix{A & B \\
          C & D }
\end{displaymath}
```



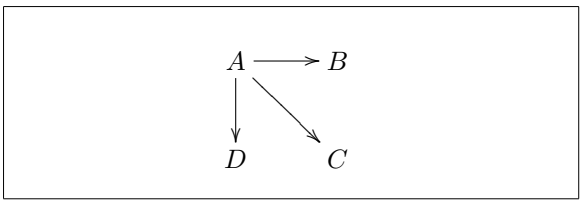
\xymatrix 命令必须在数学公式模式下使用。这里，我们定义了一个两行、两列的矩阵。我们只要使用\ar 命令加上直箭头就把这个矩阵变成图了。

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
           D \ar[u] & C \ar[l] }
\end{displaymath}
```



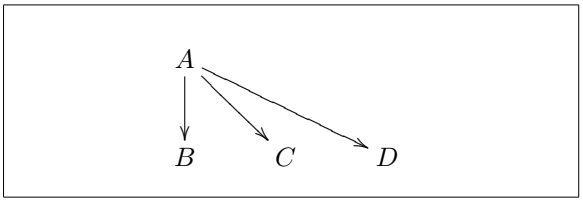
箭头命令应该放置在箭头起始的单元格，命令的参数为箭头指向的方向(上、下、右和左)。

```
\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[r] & B \\
  D                       & C }
\end{displaymath}
```



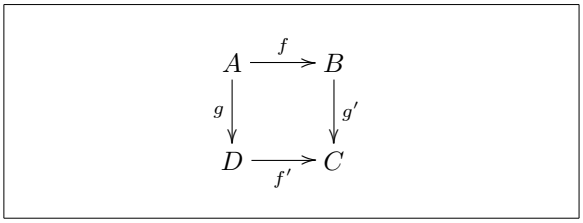
要画对角箭头只要使用多于一个的方向参数，事实上，你可以通过重复方向参数来获得不同角度的箭头。

```
\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[drr] & & \\
  B                         & C & D }
\end{displaymath}
```



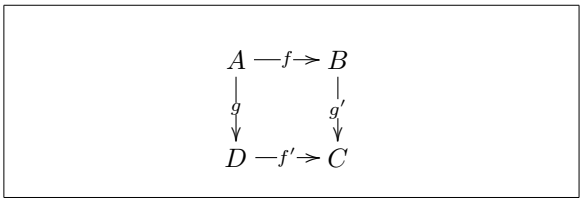
通过在箭头上加文本标签，我们可以得到更有趣的图。要做到这一点，只要使用通用的上标和下标操作符就可以了。

```
\begin{displaymath}
\xymatrix{
  A \ar[r]^f \ar[d]_g &
             B \ar[d]^{g'} \\
  D \ar[r]_{f'}       & C }
\end{displaymath}
```



如例子中所示，我们可以象在数学模式中那样使用这些操作符。唯一不同的一点是在这里上标表示“在箭头上面”，而下标表示“在箭头下面”。还有第三个操作符，那就是竖杠：| 。使用它可以把文本放置在箭头中间

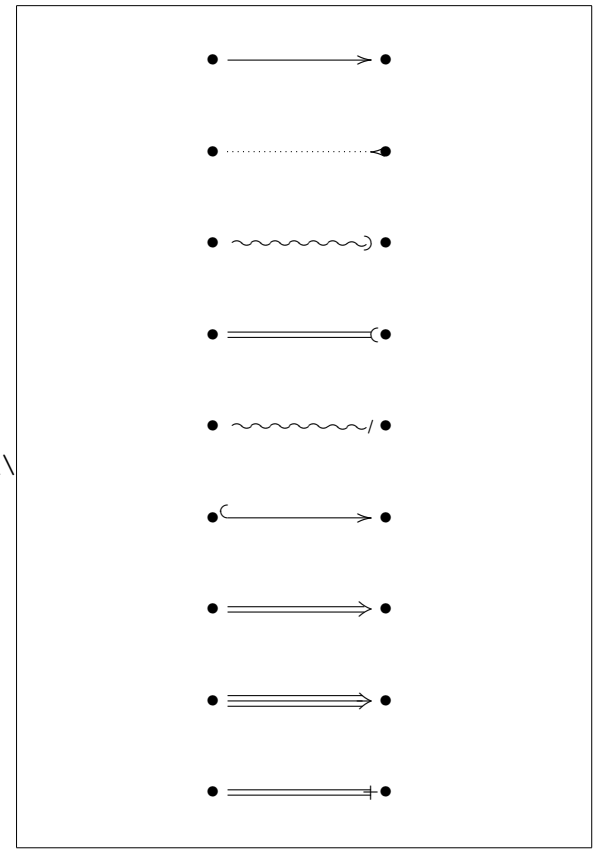
```
\begin{displaymath}
\xymatrix{
  A \ar[r]|f \ar[d]|g &
             B \ar[d]|{g'} \\
  D \ar[r]|{f'}       & C }
\end{displaymath}
```



要在箭头中间保留一个空位，请使用命令`\ar[...]|\hole` 。

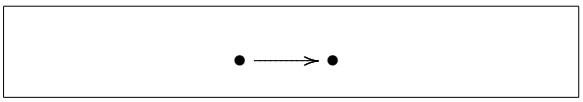
有些情况下，让不同类型的箭头有差别非常重要。这可以通过放置文本标签在箭头上做到，也可以通过命令改变箭头的外观，如下例所示：

```
\shorthandoff{"}
\begin{displaymath}
\xymatrix{
 \bullet\ar@{->}[rr] && \bullet\\
 \bullet\ar@{.<}[rr] && \bullet\\
 \bullet\ar@{~)}[rr] && \bullet\\
 \bullet\ar@{=(}[rr] && \bullet\\
 \bullet\ar@{~/}[rr]  && \bullet\\
 \bullet\ar@{^{(}->}[rr]  && \bullet\\
 \bullet\ar@2{->}[rr]  && \bullet\\
 \bullet\ar@3{->}[rr]  && \bullet\\
 \bullet\ar@{=+}[rr]   && \bullet
}
\end{displaymath}
\shorthandon{"}
```

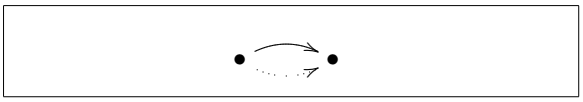


注意下面两图的差别：

```
\begin{displaymath}
\xymatrix{
 \bullet \ar[r]
         \ar@{.>}[r] &
 \bullet
}
\end{displaymath}
```



```
\begin{displaymath}
\xymatrix{
 \bullet \ar@/^/[r]
         \ar@/_/@{.>}[r] &
 \bullet
}
\end{displaymath}
```



斜杠之间的修饰符定义了画曲线的方式。Xy-pic 包提供了多种改变曲线

绘制的方式，要得到进一步的信息，请查阅Xy-pic 包的文档。

第六章

Customising LaTeX

Documents produced with the commands you have learned up to this point will look acceptable to a large audience. While they are not fancy-looking, they obey all the established rules of good typesetting, which will make them easy to read and pleasant to look at.

However, there are situations where LaTeX does not provide a command or environment that matches your needs, or the output produced by some existing command may not meet your requirements.

In this chapter, I will try to give some hints on how to teach LaTeX new tricks and how to make it produce output that looks different from what is provided by default.

6.1 New Commands, Environments and Packages

You may have noticed that all the commands I introduce in this book are typeset in a box, and that they show up in the index at the end of the book. Instead of directly using the necessary LaTeX commands to achieve this, I have created a package in which I defined new commands and environments for this purpose. Now I can simply write:

```
\begin{lscommand}
\ci{dum}
\end{lscommand}
```

In this example, I am using both a new environment called `lscommand`, which is responsible for drawing the box around the command, and a new

command named \ci, which typesets the command name and makes a corresponding entry in the index. You can check this out by looking up the \dum command in the index at the back of this book, where you'll find an entry for \dum, pointing to every page where I mentioned the \dum command.

If I ever decide that I do not like the commands to be typeset in a box any more, I can simply change the definition of the lscommand environment to create a new look. This is much easier than going through the whole document to hunt down all the places where I have used some generic LaTeX commands to draw a box around some word.

6.1.1 New Commands

To add your own commands, use the

\newcommand{*name*}[*num*]{*definition*}

command. Basically, the command requires two arguments: the *name* of the command you want to create, and the *definition* of the command. The *num* argument in square brackets is optional and specifies the number of arguments the new command takes (up to 9 are possible). If missing it defaults to 0, i.e. no argument allowed.

The following two examples should help you to get the idea. The first example defines a new command called \tnss. This is short for "The Not So Short Introduction to LaTeX 2_ε." Such a command could come in handy if you had to write the title of this book over and over again.

```
\newcommand{\tnss}{The not
    so Short Introduction to
    \LaTeXe}
This is ``\tnss'' \ldots{}
``\tnss''
```

This is "The not so Short Introduction to LaTeX 2_ε" ... "The not so Short Introduction to LaTeX 2_ε"

The next example illustrates how to define a new command that takes one argument. The #1 tag gets replaced by the argument you specify. If you wanted to use more than one argument, use #2 and so on.

```
\newcommand{\txsit}[1]
 {This is the \emph{#1} Short
      Introduction to \LaTeXe}
% in the document body:
\begin{itemize}
\item \txsit{not so}
\item \txsit{very}
\end{itemize}
```

- This is the *not so* Short Introduction to LaTeX 2ε
- This is the *very* Short Introduction to LaTeX 2ε

LaTeX will not allow you to create a new command that would overwrite an existing one. But there is a special command in case you explicitly want this: `\renewcommand`. It uses the same syntax as the `\newcommand` command.

In certain cases you might also want to use the `\providecommand` command. It works like `\newcommand`, but if the command is already defined, LaTeX 2ε will silently ignore it.

There are some points to note about whitespace following LaTeX commands. See page 6 for more information.

6.1.2 New Environments

Just as with the `\newcommand` command, there is a command to create your own environments. The `\newenvironment` command uses the following syntax:

`\newenvironment{`*name*`}[`*num*`]{`*before*`}{`*after*`}`

Again `\newenvironment` can have an optional argument. The material specified in the *before* argument is processed before the text in the environment gets processed. The material in the *after* argument gets processed when the `\end{`*name*`}` command is encountered.

The example below illustrates the usage of the `\newenvironment` command.

```
\newenvironment{king}
 {\rule{1ex}{1ex}%
      \hspace{\stretch{1}}}
 {\hspace{\stretch{1}}%
      \rule{1ex}{1ex}}

\begin{king}
My humble subjects \ldots
\end{king}
```

■ My humble subjects . . . ■

The *num* argument is used the same way as in the `\newcommand` command. LaTeX makes sure that you do not define an environment that already exists. If you ever want to change an existing command, you can use the `\renewenvironment` command. It uses the same syntax as the `\newenvironment` command.

The commands used in this example will be explained later. For the `\rule` command see page 127, for `\stretch` go to page 120, and more information on `\hspace` can be found on page 120.

6.1.3 Your Own Package

If you define a lot of new environments and commands, the preamble of your document will get quite long. In this situation, it is a good idea to create a LaTeX package containing all your command and environment definitions. You can then use the `\usepackage` command to make the package available in your document.

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{The not so Short Introduction to \LaTeXe}
\newcommand{\txsit}[1]{The \emph{#1} Short
                       Introduction to \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

图 6.1: Example Package.

Writing a package basically consists of copying the contents of your document preamble into a separate file with a name ending in `.sty`. There

is one special command,

`\ProvidesPackage{`*package name*`}`

for use at the very beginning of your package file. `\ProvidesPackage` tells LaTeX the name of the package and will allow it to issue a sensible error message when you try to include a package twice. Figure 6.1 shows a small example package that contains the commands defined in the examples above.

6.2 Fonts and Sizes

6.2.1 Font Changing Commands

LaTeX chooses the appropriate font and font size based on the logical structure of the document (sections, footnotes, ...). In some cases, one might like to change fonts and sizes by hand. To do this, you can use the commands listed in Tables 6.1 and 6.2. The actual size of each font is a design issue and depends on the document class and its options. Table 6.3 shows the absolute point size for these commands as implemented in the standard document classes.

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

The small and **bold** Romans ruled all of great big *Italy*.

One important feature of LaTeX 2_ε is that the font attributes are independent. This means that you can issue size or even font changing commands, and still keep the bold or slant attribute set earlier.

In *math mode* you can use the font changing *commands* to temporarily exit *math mode* and enter some normal text. If you want to switch to another font for math typesetting you need another special set of commands; refer to Table 6.4.

In connection with the font size commands, curly braces play a significant role. They are used to build *groups*. Groups limit the scope of most LaTeX commands.

```
He likes {\LARGE large and
{\small small} letters}.
```

He likes large and small letters.

表 6.1: Fonts.

`\textrm{...}`	roman	`\textsf{...}`	sans serif
`\texttt{...}`	`typewriter`		
`\textmd{...}`	medium	`\textbf{...}`	**bold face**
`\textup{...}`	upright	`\textit{...}`	*italic*
`\textsl{...}`	*slanted*	`\textsc{...}`	SMALL CAPS
`\emph{...}`	*emphasized*	`\textnormal{...}`	document font

表 6.2: Font Sizes.

`\tiny`	tiny font	`\Large`	larger font
`\scriptsize`	very small font	`\LARGE`	very large font
`\footnotesize`	quite small font	`\huge`	huge
`\small`	small font	`\Huge`	largest
`\normalsize`	normal font		
`\large`	large font		

表 6.3: Absolute Point Sizes in Standard Classes.

size	10pt (default)	11pt option	12pt option
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

The font size commands also change the line spacing, but only if the paragraph ends within the scope of the font size command. The closing curly brace } should therefore not come too early. Note the position of the \par command in the next two examples. [1]

```
{\Large Don't read this! It is not
true. You can believe me!\par}
```

{Large Don't read this! It is not true. You can believe me!}

```
{\Large This is not true either.
But remember I am a liar.}\par
```

{Large This is not true either. But remember I am a liar.}

If you want to activate a size changing command for a whole paragraph of text or even more, you might want to use the environment syntax for font changing commands.

```
\begin{Large}
This is not true.
But then again, what is these
days \ldots
\end{Large}
```

{Large This is not true. But then again, what is these days ...}

This will save you from counting lots of curly braces.

[1] \par is equivalent to a blank line

表 6.4: Math Fonts.

Command	*Example*	*Output*
\mathcal{...}	$\mathcal{B}=c$	$\mathcal{B} = c$
\mathrm{...}	K_2	K_2
\mathbf{...}	$\sum x=\mathbf{v}$	$\sum x = \mathbf{v}$
\mathsf{...}	$\mathsf{G\times R}$	$\mathsf{G} \times \mathsf{R}$
\mathtt{...}	$\mathtt{L}(b,c)$	$\mathtt{L}(b, c)$
\mathnormal{...}	$\mathnormal{R_{19}}\neq R_{19}$	$R_{19} \neq R_{19}$
\mathit{...}	$\mathit{ffi}\neq ffi$	$\mathit{ffi} \neq ffi$

6.2.2 Danger, Will Robinson, Danger

As noted at the beginning of this chapter, it is dangerous to clutter your document with explicit commands like this, because they work in opposition to the basic idea of LaTeX, which is to separate the logical and visual markup of your document. This means that if you use the same font changing command in several places in order to typeset a special kind of information, you should use `\newcommand` to define a "logical wrapper command" for the font changing command.

```
\newcommand{\oops}[1]{\textbf{#1}}
Do not \oops{enter} this room,
it's occupied by a \oops{machine}
of unknown origin and purpose.
```

Do not **enter** this room, it's occupied by a **machine** of unknown origin and purpose.

This approach has the advantage that you can decide at some later stage that you want to use some visual representation of danger other than `\textbf`, without having to wade through your document, identifying all the occurrences of `\textbf` and then figuring out for each one whether it was used for pointing out danger or for some other reason.

6.2.3 Advice

To conclude this journey into the land of fonts and font sizes, here is a little word of advice:

Remember! *The* MO RE fonts you use **in** a `document`, *the* more READABLE and ***beautiful*** *it becomeS*.

6.3 Spacing

6.3.1 Line Spacing

If you want to use larger inter-line spacing in a document, you can change its value by putting the

`\linespread{`*factor*`}`

command into the preamble of your document. Use `\linespread{1.3}` for "one and a half" line spacing, and `\linespread{1.6}` for "double" line spa-

cing. Normally the lines are not spread, so the default line spread factor is 1.

Note that the effect of the `\linespread` command is rather drastic and not appropriate for published work. So if you have a good reason for changing the line spacing you might want to use the command:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
           {1.5\baselineskip}
This paragraph is typeset with the
baseline skip set to 1.5 of what
it was before. Note the par command
at the end of the paragraph.\par}

This paragraph has a clear purpose,
it show that after the curly brace
has been closed everything is back
to normal.
```

This paragraph is typeset with the baseline skip set to 1.5 of what it was before. Note the par command at the end of the paragraph.

This paragraph has a clear purpose, it show that after the curly brace has been closed everything is back to normal.

6.3.2 Paragraph Formatting

In LaTeX, there are two parameters influencing paragraph layout. By placing a definition like

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

in the preamble of the input file, you can change the layout of paragraphs. These two commands increase the space between two paragraphs while setting the paragraph indent to zero.

The `plus` and `minus` parts of the length above tell TeX that it can compress and expand the inter paragraph skip by the amount specified, if this is necessary to properly fit the paragraphs onto the page.

In continental Europe, paragraphs are often separated by some space and not indented. But beware, this also has its effect on the table of contents. Its lines get spaced more loosely now as well. To avoid this, you might want

to move the two commands from the preamble into your document to some place after the `\tableofcontents` or to not use them at all, because you'll find that most professional books use indenting and not spacing to separate paragraphs.

If you want to indent a paragraph that is not indented, you can use

`\indent`

at the beginning of the paragraph.[2] Obviously, this will only have an effect when `\parindent` is not set to zero.

To create a non-indented paragraph, you can use

`\noindent`

as the first command of the paragraph. This might come in handy when you start a document with body text and not with a sectioning command.

6.3.3 Horizontal Space

LaTeX determines the spaces between words and sentences automatically. To add horizontal space, use:

`\hspace{`*length*`}`

If such a space should be kept even if it falls at the end or the start of a line, use `\hspace*` instead of `\hspace`. The *length* in the simplest case is just a number plus a unit. The most important units are listed in Table 6.5.

```
This\hspace{1.5cm}is a space
of 1.5 cm.
```

This is a space of 1.5 cm.

The command

`\stretch{`*n*`}`

generates a special rubber space. It stretches until all the remaining space on a line is filled up. If two `\hspace{\stretch{`*n*`}}` commands are issued on the same line, they grow according to the stretch factor.

[2] To indent the first paragraph after each section head, use the indentfirst package in the 'tools' bundle.

表 6.5: TEX Units.

`mm`	millimetre ≈ 1/25 inch
`cm`	centimetre = 10 mm
`in`	inch = 25.4 mm
`pt`	point ≈ 1/72 inch ≈ $\frac{1}{3}$ mm
`em`	approx width of an 'M' in the current font
`ex`	approx height of an 'x' in the current font

```
x\hspace{\stretch{1}}
x\hspace{\stretch{3}}x
```

x x x

When using horizontal space together with text, it may make sense to make the space adjust its size relative to the size of the current font. This can be done by using the text-relative units `em` and `ex`:

```
{\Large{}big\hspace{1em}y}\\
{\tiny{}tin\hspace{1em}y}
```

big y
tin y

6.3.4 Vertical Space

The space between paragraphs, sections, subsections, ... is determined automatically by LATEX. If necessary, additional vertical space *between two paragraphs* can be added with the command:

`\vspace{`*length*`}`

This command should normally be used between two empty lines. If the space should be preserved at the top or at the bottom of a page, use the starred version of the command, `\vspace*`, instead of `\vspace`.

The `\stretch` command, in connection with `\pagebreak`, can be used to typeset text on the last line of a page, or to centre text vertically on a page.

```
Some text \ldots
```

```
\vspace{\stretch{1}}
This goes onto the last line of the page.\pagebreak
```

Additional space between two lines of *the same* paragraph or within a table is specified with the

`\\[`*length*`]`

command.

With `\bigskip` and `\smallskip` you can skip a predefined amount of vertical space without having to worry about exact numbers.

6.4 Page Layout

LaTeX 2_ε allows you to specify the paper size in the `\documentclass` command. It then automatically picks the right text margins, but sometimes you may not be happy with the predefined values. Naturally, you can change them. Figure 6.2 shows all the parameters that can be changed. The figure was produced with the layout package from the tools bundle.[3]

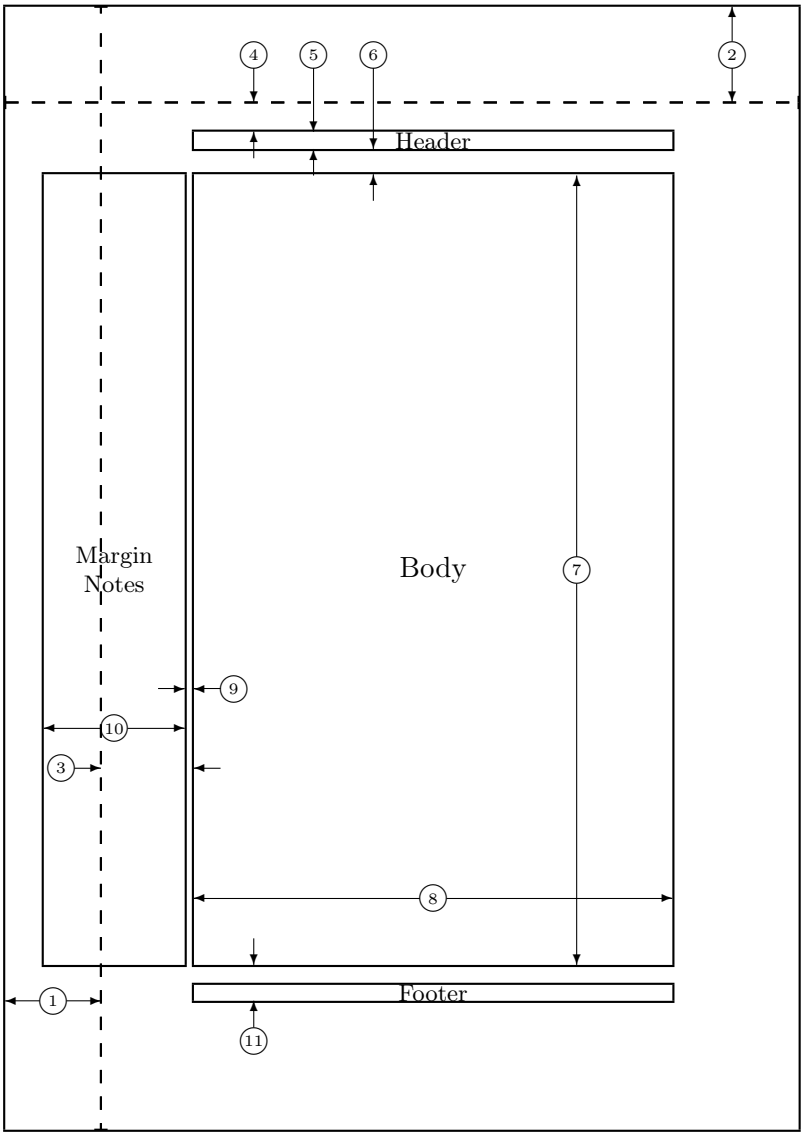
WAIT! . . . before you launch into a "Let's make that narrow page a bit wider" frenzy, take a few seconds to think. As with most things in LaTeX, there is a good reason for the page layout to be as it is.

Sure, compared to your off-the-shelf MS Word page, it looks awfully narrow. But take a look at your favourite book[4] and count the number of characters on a standard text line. You will find that there are no more than about 66 characters on each line. Now do the same on your LaTeX page. You will find that there are also about 66 characters per line. Experience shows that the reading gets difficult as soon as there are more characters on a single line. This is because it is difficult for the eyes to move from the end of one line to the start of the next one. This is also why newspapers are typeset in multiple columns.

So if you increase the width of your body text, keep in mind that you are making life difficult for the readers of your paper. But enough of the cautioning, I promised to tell you how you do it . . .

[3] `macros/latex/required/tools`

[4] I mean a real printed book produced by a reputable publisher.



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 22pt or \evensidemargin	4	\topmargin = 22pt
5	\headheight = 13pt	6	\headsep = 19pt
7	\textheight = 595pt	8	\textwidth = 360pt
9	\marginparsep = 7pt	10	\marginparwidth = 106pt \marginparpush = 5pt (not shown)
11	\footskip = 27pt \hoffset = 0pt \paperwidth = 597pt		\voffset = 0pt \paperheight = 845pt

图 6.2: Page Layout Parameters.

LaTeX provides two commands to change these parameters. They are usually used in the document preamble.

The first command assigns a fixed value to any of the parameters:

`\setlength{`*parameter*`}{`*length*`}`

The second command adds a length to any of the parameters:

`\addtolength{`*parameter*`}{`*length*`}`

This second command is actually more useful than the `\setlength` command, because you can now work relative to the existing settings. To add one centimetre to the overall text width, I put the following commands into the document preamble:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

In this context, you might want to look at the calc package. It allows you to use arithmetic operations in the argument of `\setlength` and other places where you can enter numeric values into function arguments.

6.5 More Fun With Lengths

Whenever possible, I avoid using absolute lengths in LaTeX documents. I rather try to base things on the width or height of other page elements. For the width of a figure this could be `\textwidth` in order to make it fill the page.

The following 3 commands allow you to determine the width, height and depth of a text string.

`\settoheight{`*variable*`}{`*text*`}`
`\settodepth{`*variable*`}{`*text*`}`
`\settowidth{`*variable*`}{`*text*`}`

The example below shows a possible application of these commands.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Where}$a$,
$b$ -- are adjoin to the right
angle of a right-angled triangle.

$c$ -- is the hypotenuse of
the triangle and feels lonely.

$d$ -- finally does not show up
here at all. Isn't that puzzling?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Where: a, b – are adjoin to the right angle of a right-angled triangle.

c – is the hypotenuse of the triangle and feels lonely.

d – finally does not show up here at all. Isn't that puzzling?

6.6 Boxes

LaTeX builds up its pages by pushing around boxes. At first, each letter is a little box, which is then glued to other letters to form words. These are again glued to other words, but with special glue, which is elastic so that a series of words can be squeezed or stretched as to exactly fill a line on the page.

I admit, this is a very simplistic version of what really happens, but the point is that TeX operates on glue and boxes. Letters are not the only things that can be boxes. You can put virtually everything into a box, including other boxes. Each box will then be handled by LaTeX as if it were a single letter.

In the past chapters you have already encountered some boxes, although I did not tell you. The `tabular` environment and the `\includegraphics`, for example, both produce a box. This means that you can easily arrange two tables or images side by side. You just have to make sure that their combined width is not larger than the textwidth.

You can also pack a paragraph of your choice into a box with either the

`\parbox[`*pos*`]{`*width*`}{`*text*`}`

command or the

`\begin{minipage}[`*pos*`]{`*width*`}` text `\end{minipage}`

environment. The `pos` parameter can take one of the letters `c, t` or `b` to control the vertical alignment of the box, relative to the baseline of the surrounding text. `width` takes a length argument specifying the width of the box. The main difference between a `minipage` and a `\parbox` is that you cannot use all commands and environments inside a `parbox`, while almost anything is possible in a `minipage`.

While `\parbox` packs up a whole paragraph doing line breaking and everything, there is also a class of boxing commands that operates only on horizontally aligned material. We already know one of them; it's called `\mbox`. It simply packs up a series of boxes into another one, and can be used to prevent LaTeX from breaking two words. As you can put boxes inside boxes, these horizontal box packers give you ultimate flexibility.

`\makebox[`*width*`][`*pos*`]{`*text*`}`

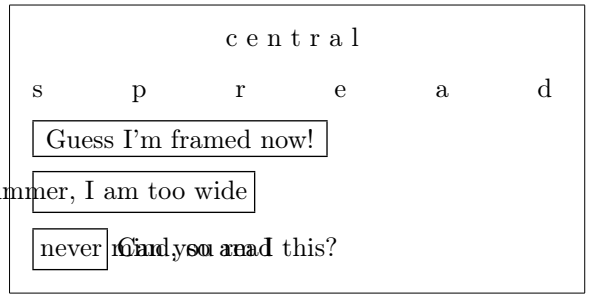
`width` defines the width of the resulting box as seen from the outside.[5] Besides the length expressions, you can also use `\width`, `\height`, `\depth`, and `\totalheight` in the width parameter. They are set from values obtained by measuring the typeset *text*. The *pos* parameter takes a one letter value: **c**enter, **l**eft flush, **r**ight, flush, or **s**, which spreads the text inside the box to fill it.

The command `\framebox` works exactly the same as `\makebox`, but it draws a box around the text.

The following example shows you some things you could do with the `\makebox` and `\framebox` commands.

[5] This means it can be smaller than the material inside the box. You can even set the width to 0pt so that the text inside the box will be typeset without influencing the surrounding boxes.

```
\makebox[\textwidth]{%
    c e n t r a l}\par
\makebox[\textwidth][s]{%
    s p r e a d}\par
\framebox[1.1\width]{Guess I'm
    framed now!} \par
\framebox[0.8\width][r]{Bummer,
    I am too wide} \par
\framebox[1cm][l]{never
    mind, so am I}
Can you read this?
```

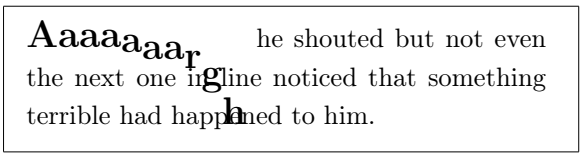


Now that we control the horizontal, the obvious next step is to go for the vertical.[6] No problem for LaTeX. The

`\raisebox{`*lift*`}[`*depth*`][`*height*`]{`*text*`}`

command lets you define the vertical properties of a box. You can use `\width`, `\height`, `\depth`, and `\totalheight` in the first three parameters, in order to act upon the size of the box inside the *text* argument.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}
he shouted but not even the next
one in line noticed that something
terrible had happened to him.
```



6.7 Rules and Struts

A few pages back you may have noticed the command

`\rule[`*lift*`]{`*width*`}{`*height*`}`

In normal use it produces a simple black box.

[6] Total control is only to be obtained by controlling both the horizontal and the vertical ...

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```

This is useful for drawing vertical and horizontal lines. The line on the title page, for example, has been created with a `\rule` command.

A special case is a rule with no width but a certain height. In professional typesetting, this is called a strut. It is used to guarantee that an element on a page has a certain minimal height. You could use it in a `tabular` environment to make sure a row has a certain minimum height.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\\
\hline
\rule{0pt}{4ex}Strut\\
\hline
\end{tabular}
```

The End.

参考文献

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The LaTeX Graphics Companion.* Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.

[5] 每个 LaTeX 安装都应该提供一个称作 *LaTeX Local Guide* 的文档, 它描述了对于本系统比较特别的地方。它通常包含在 `local.tex` 文件中。不幸的是, 一些比较懒惰的管理员没有提供这个文档。在这种情况下，你可以寻求当地 LaTeX 高手的帮助。

[6] LaTeX3 Project Team. *LaTeX 2ε for authors.* 包含在 LaTeX 2ε 发行中，文件名通常是 `usrguide.tex`。

[7] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers.* 包含在 LaTeX 2ε 发行中，文件名通常是 `clsguide.tex`。

[8] LaTeX3 Project Team. *LaTeX 2ε Font selection.* 包含在 LaTeX 2ε 发行中，文件名通常是 `fntguide.tex`.

[9] D. P. Carlisle. *Packages in the 'graphics' bundle.* 包含在 'graphics' 系列宏包中，文件名通常是 `grfguide.tex`，可以从你获得 LaTeX 发行的地方得到。

[10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of LaTeX's verbatim Environments.* 包含在 'tools' 系列宏包中，文件名通常是 `verbatim.dtx`，可以从你获得 LaTeX 发行的地方得到。

[11] Vladimir Volovich, Werner Lemberg and LaTeX3 Project Team. *Cyrillic languages support in LaTeX.* 包含在 LaTeX 2_ε 发行中，文件名通常是 `cyrguide.tex`.

[12] Graham Williams. *The TeX Catalogue* 是一个关于许多 TeX 和 LaTeX 相关宏包的非常全面的列表。可以在线从 CTAN:/help/Catalogue/catalogue.html 获得。

[13] Keith Reckdahl. *Using EPS Graphics in LaTeX 2_ε Documents*，描述了关于 EPS 文件以及在 LaTeX 文档中使用它的一切、甚至还有很多你从未想过的内容。可以在线从 CTAN:/info/epslatex.ps 获得。

[14] Kristoffer H. Rose. *XY-pic User's Guide.* 可以从 CTAN 与 XY-pic 发行一起下载。

[15] John D. Hobby. *A User's Manual for MetaPost.* 可以从 http://cm.bell-labs.com/who/hobby/ 下载。

[16] Alan Hoenig. *TeX Unbound.* Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)

[17] Urs Oswald. *Graphics in LaTeX 2_ε*，包含一些在 `picture` 环境中生成任意圆和椭圆的 Java 源代码，以及 *MetaPost - A Tutorial*。它们都可以从 http://www.ursoswald.ch 下载。